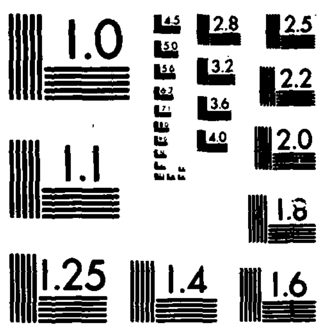
AD-A163 819 A MODEL OF AIRFLOW AND DIFFUSION IN COMPLEX TERRAIN (MADICT)(U) SRI INTERNATIONAL MENLO PARK CA F L LUDWIG NOV 85 ARO-19630.5-GS DAAG29-83-K-0009 1/2

UNCLASSIFIED F/G 4/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

APPROVED FOR PUBLIC RELEASE; ARO 19630.5-6
DISTRIBUTION UNLIMITED

(2)

AD-A163 819

USERS' GUIDE FOR A MODEL OF AIRFLOW AND DIFFUSION IN COMPLEX TERRAIN (MADICT)

Technical Report 1

November 1985

DTIC
ELECTE
FEB 1 0 1986
S D
D

Prepared by:

F. L. Ludwig
Atmospheric Science Center

Prepared for:

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709
Attn: Dr. Walter Bach
Scientific Program Officer

ARO Contract No. DAAG29-83-K-0009

SRI Project 5047

DTIC FILE COPY

SRI International
333 Ravenswood Avenue
Menlo Park, California 94025-3493
415/326-6200

SRI International ®

THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION, UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.

MASTER COPY - FOR REPRODUCTION PURPOSES

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD-A163819

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. N/A	3. RECIPIENT'S CATALOG NUMBER N/A
4. TITLE (and Subtitle) USERS' GUIDE FOR A MODEL OF AIRFLOW AND DIFFUSION IN COMPLEX TERRAIN (MADICT)		5. TYPE OF REPORT & PERIOD COVERED Users' Guide/Technical
		6. PERFORMING ORG. REPORT NUMBER 5047
7. AUTHOR(s) F. L. Ludwig		8. CONTRACT OR GRANT NUMBER(s) DAAG29-83-K-0009
9. PERFORMING ORGANIZATION NAME AND ADDRESS SRI International 333 Ravenswood Avenue Menlo Park, California 94025		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS N/A
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office Post Office Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE November 1985
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Geosciences Division U. S. Army Research Office P. O. Box 12211 Research Triangle Park, NC 27709		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

NA

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A model of transport and diffusion during time- and space-varying meteorological conditions is described, along with relevant supporting theory. The model calculates 3-dimensional wind fields in complex terrain from limited observational input. Superposition of previously calculated mass-conserving wind field solutions is used to keep the required calculations to a minimum. Limited discussions of the required wind calculations are included, along with listings of applicable computer codes. The transport and diffusion module uses "puffs"

-- Continued on back --

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

to simulate continuous plume behavior. Techniques are described for minimizing the required calculations. The report includes complete user instructions, a FORTRAN77 computer code, and two sample problems for the transport and diffusion model.

USERS' GUIDE FOR A MODEL OF AIRFLOW AND DIFFUSION IN COMPLEX TERRAIN (MADICT)

Technical Report 1

November 1985

By:

F. L. Ludwig
Atmospheric Science Center
SRI International

Prepared for:

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709
Scientific Program Officer: Dr. Walter Bach

ARO Contract No. DAAG29-83-K-0009

SRI Project 5047

Approved by:

Warren B. Johnson, Director
Atmospheric Science Center

David D. Elliott, Vice President
Research and Analysis Division

SRI International
333 Ravenswood Avenue, Menlo Park, California 94025
415/326-6200

FOREWORD

The author wishes to thank Kenneth C. Nitz and Roy M. Endlich for their assistance in the wind field modeling aspects of this work. The efforts of Valerie Ramsay, Dorothy Sevela, and Jane Cano in the production of this document is also gratefully acknowledged. The preparation of this User's Manual was supported solely by the U.S. Army Research Office under Contract No. DAAG29-83-K-0009.

CONTENTS

ILLUSTRATIONS

Tables

I INTRODUCTION

The Model of Airflow and Diffusion In Complex Terrain (MADICT) addresses two somewhat contradictory modeling requirements. First, it requires relatively modest computational resources so that on-line real-time applications in the field are feasible. Second, MADICT can be used under conditions where airflow changes in both time and space. Some of the physical processes governing spatially varying winds and complex topography have been included, as have the responses of smokes and gases to those winds and other meteorological conditions.

In order to achieve the two goals discussed above, it has been necessary to be selective about the processes that are included in the model. For example, the principle of conservation of mass has been included in the determination of airflow in complex terrain because that seems to be a very important determinant of such flow. A Lagrangian formulation of the model allows us to treat larger scale wind variations, but not those whose size is comparable to that of the smoke plume itself. Again, the most important effects are included in the model, but other processes which are sometimes important (e.g., the effects of strong vertical wind shear) have not been included.

MADICT draws heavily on other computer programs and existing modeling concepts. For example, many of the efficiency-improving ideas in the transport and diffusion model had their genesis in an on-line plume dispersion model for a power plant (Ludwig, Gasiorek, and Ruff, 1977). The methods for estimating airflow in complex terrain are derived from techniques that were used to estimate wind energy potential in such areas (e.g., Bhumralkar, et al, 1980; Ludwig and Byrd, 1980). The separate origins of different parts of the model have resulted in a program with reasonably distinct modules. This should make it easier for the user to introduce modifications that suit his or her needs. The model presented here is written in FORTRAN77; a somewhat simpler (and less structured) version has also been written in BASIC for use on microprocessor based systems (Ludwig, 1984).

We have tried to make the model efficient by reducing the amount of computation and internal memory that are required. Efficiency has been achieved in two ways. First, the computational requirements of the problem have been carefully analyzed to determine what calculations are required. For example, no attempt is made to calculate the contribution from some part of a smoke plume that is far from a receptor. A minimum number of elements is used to represent a smoke plume, and some calculations can be made ahead of time and stored for later use. In the latter category, the generation of wind fields in complex terrain is done in two steps; the part of a computation that requires the most effort is done once with the results stored as a data file which is used as input for much simpler arithmetic calculations during the running of the model.

The remainder of this report discusses the theory of the model and the techniques used to reduce the required calculations. The structure of the model is also discussed so that users can modify the code for a particular

application. Instructions are provided on model usage, and the application of the associated preprocessing programs. These instructions include descriptions of the required inputs and the outputs. The chapter providing operational instructions is intended to stand alone, although the user should read the other chapters to get a better basis for interpreting the results.

The report concludes with several appendices that list the program code for both the model and the preprocessers. The final appendix provides two sample problems.

II DESCRIPTION OF THE MODEL THEORY

A. Background

The model has two major parts: (1) a module that describes the winds (in three dimensions) throughout the modeling domain, and how those winds change with time; and (2) a module that deals with the transport and diffusion of materials emitted into the atmosphere that accounts for how the winds and the prevailing turbulence affect motion and dilution of the smokes or other emitted materials. The meteorological module of MADICT emphasizes the description of the spatial variability of the winds. Observed winds are interpolated and adjusted to account for some of the effects of complex topography. The process invokes the principle of mass conservation which adjusts the winds to achieve nondivergence. Temporal variability is introduced by changing wind fields in response to changed wind observations. Real-time calculations are reduced by using the linear properties of the problem to form any arbitrary solution by superposition of stored, standard wind field solutions that have been obtained ahead of time using the complete computational procedure. Interpolation of the coefficients used to form the linear wind field combinations allows winds to be interpolated between observation times.

The model only accounts for vertical variations in atmospheric stability and their changes with time, and this is only treated in a very rudimentary fashion. Lateral variability in the diffusive properties of the atmosphere is not addressed. In principle, it would not be particularly difficult to deal with spatially varying diffusion, if the necessary information were available.

This model approximates the continuous emissions from the source with a series of discreet "puffs" that move with the wind at their centers. The puffs grow and become more dilute in response to turbulent diffusion. Computational efficiency is achieved by minimizing the number of puffs that are used through the merging of puffs that overlap and the discarding of those outside the domain. The puff modeling approach does a good job of simulating plume behavior in spatially varying wind fields that do not change significantly over distances comparable to the size of the puffs. This is an improvement over steady-state modeling approaches, but it is inaccurate when there is strong wind shear and should be borne in mind when applying the model and interpreting the results.

The model is intended to be applied to nonbuoyant smoke sources, so it includes no plume rise module. This may be a serious omission for some applications.

MADICT is intended for real-time use in the field, so good graphical displays would be very helpful for interpreting results. However, the differences among graphical display systems are so great that no special displays have been developed for this application. The original microcomputer version (Ludwig, 1984) included a very simple, character-based display. The FORTRAN77 version presented here uses graphics subroutines available from the National

Center for A*mospheric Research (NCAR) to display concentration patterns. Other graphics subroutines could be substituted.

B. Wind Field Calculations

The method by which wind fields are generated has evolved from the work of Sherman (1978), as modified by Bhumralkar, et al (1980). The suggestions made by Ludwig and Byrd (1980) have been implemented in order to reduce online calculations. Inasmuch as the full model is not used in MADICT, it is not essential that any particular wind model be used. The model discussed here has been described by Endlich (1984), and portions of the users' instructions prepared by Endlich and Lee (1983) are included in this document. Whatever wind generation model is used, it must have the following characteristics:

1. It must produce three-dimensional, nondivergent wind fields that account for variations in the underlying topography.
2. The output winds must be available at the top and bottom of the modeled layer over a uniform grid.
3. For any specified boundary layer thickness, the output wind field must be linearly related to the input wind observations.
4. The locations from which the input wind observations are obtained should remain the same from case to case.
5. The heights of the surface and the top of the modeled layer above the surface must be output for each grid point.

In addition to the above criteria, it is also desirable to have historical wind data from the observation sites so that those patterns which account for the largest part of the variance in the data can be identified. As will be discussed later, this allows calculations to be simplified and provides some smoothing of smaller-scale features of the wind fields. This latter approach uses the u and v wind components from each site at 3-hour intervals, along with the geostrophic wind for the area. A computer program computes the covariance matrix for all the input wind components (including the geostrophic wind) and outputs the eigenvectors of that matrix. For n reporting stations and one geostrophic wind there are 2(n+1) eigenvectors and one mean vector (Ludwig and Byrd, 1980). The components of the mean vector are the inputs which would be used for the model if each observing station reported its long term average wind, along with the average geostrophic wind.

The first eigenvector corresponds to a set of input data that accounts for the largest part of the variance in the input data set. In essence, each eigenvector can be considered as a set of input data that is linearly independent of all the other eigenvector input data sets. If the appropriate conditions are met (see Ludwig and Byrd, 1980), then solutions for any given set of inputs can be decomposed to a linear combination of the eigenvectors and the mean. Although, any linearly independent set of inputs can be used, the eigenvectors have some advantages that are discussed below.

Figure 1 shows the observed wind components as a sum of their corresponding means (the column vector at the right) and multiples of the set of normalized eigenvectors. The coefficients of the linear combination are symbolized with the constants, a_1 through a_8. These coefficients are the inner products of the input data vector (at the left of Figure 2) and each of the independent column vectors (the eigenvectors in Figure 2). These inner products are calculated in the subroutine INWIND., using the following equation:

$$a_j = \sum_{k=1}^{N} \left[\left(U_k - \bar{U}_k\right) u_{jk} + \left(V_k - \bar{V}_k\right) v_{jk} \right]$$

where a_j is the coefficient for the jth column, and U_k and V_k are the U and V components for the kth station in the input data set. Averages of U and V are indicated by overbars; the averages are calculated for the same data set from which the eigenvectors were derived. The terms u_{jk} and v_{jk} are kth pair of wind components in the jth column vector, and N is the number of observation sites (including a geostrophic wind) from which data are available. There are two 2N components in each eigenvector.

Appendix A provides listings of programs that are used to process wind data. The program XFORM will read data for one year (at 3-hr intervals) in subroutine SUB1. That subroutine also uses a standard International Mathematical and Statistical Library (IMSL) routine (BECOVM) to calculate the covariance matrix for the input data. Another IMSL routine (EIGRS) is used to determine the eigenvectors of that covariance matrix. The means and the eigenvectors are stored for use with the full wind field generation model.

The theory of the wind field model has been described by Endlich et al (1982), Endlich and Lee (1983), and Endlich (1984). Those references should be consulted for details. A brief discussion of mass-consistent wind field models follows.

Most mass-conserving wind interpolation schemes, begin with an "initial guess" that is adjusted to remove divergence. Winds in the lowest layer are estimated for each grid point by taking a weighted vector average of the nearest observed winds. The weighting is inversely proportional to the distance between observation and grid point (Sherman, 1978; Endlich et al, 1982). An initial guess for winds at higher altitudes is more difficult to obtain. Sherman (1978) used the shape of whatever measured wind profile was available to extrapolate upward throughout the region. When no observed profile was available, she assumed a linear variation of wind with height. The models of Endlich et al (1982) and Endlich and Lee (1983) assume a uniform wind at the top of the domain, derived from the surface pressure and temperature fields, using the geostrophic and thermal wind relationships (Holmboe, 1948) to provide the upper level wind estimate. Winds at intermediate levels are derived

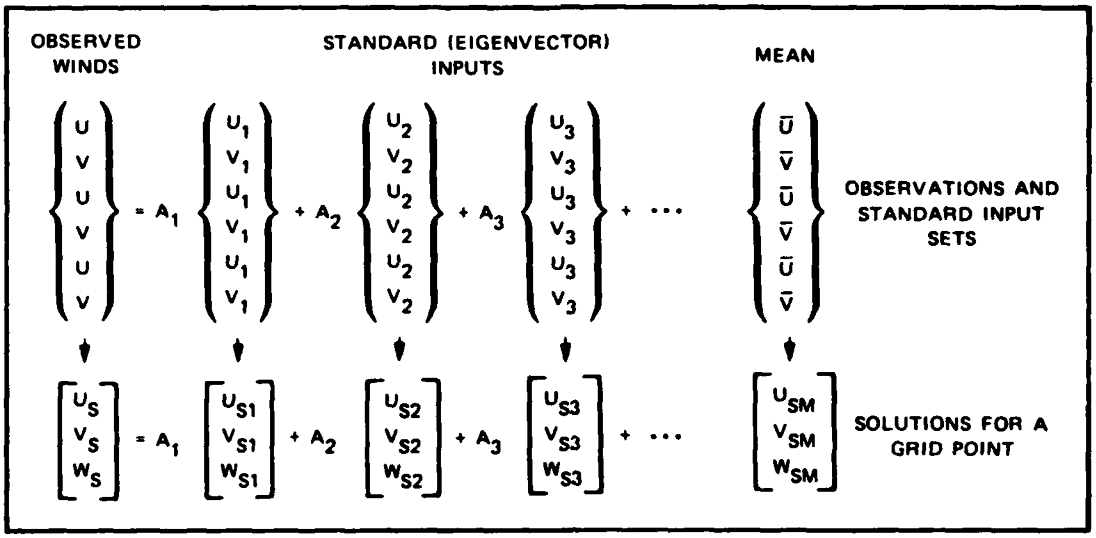


FIGURE 1 SCHEMATIC DIAGRAM OF THE METHOD OF SUPERPOSITION USED TO OBTAIN WIND FIELDS

from a log-linear interpolation between the surface wind estimate and that for the top of the domain.

Two different coordinate systems have been used for mass-consistent wind models. For example, Sherman (1978) uses standard rectangular coordinates. While Endlich et al (1980) use a curvilinear coordinate system that depends on the shape of the upper and lower boundaries of the domain. The main features of these two modeling approaches are discussed below.

Sherman (1978) sought to "minimize the variance of the difference between the adjusted values and the original values subject to the strong constraint that the 3-dimensional analyzed wind field is nondivergent," using Sasaki's (1958, 1970a,b) analytical approach to obtain a solution that minimizes the following function (integrated over the model domain):

$$S(u,v,w,\lambda) = \iiint \Big[\alpha_1^2(u - u_o)^2 + \alpha_1^2(v - v_o)^2 + \alpha_2^2(w - w_o)^2 + \lambda\nabla\cdot\vec{V}\Big]\,dxdydz \tag{1}$$

where α_1 and α_2 are weighting factors that are, in essence, used to inhibit adjustments of the vertical velocity relative to the horizontal components. Sherman suggests that the magnitude of α_2 should be about 100 times more than that of α_1. The other variables in Equation (1) are a Lagrange multiplier, $\lambda(x,y,z)$, the observed wind components with subscript "o", and the adjusted components of the wind $\vec{V}$.

The Euler-Lagrange equations whose solution will minimize Equation (1) are:

$$u = u_o + \frac{1}{2\alpha_1^2}\frac{\partial\lambda}{\partial x} \tag{2}$$

$$v = v_o + \frac{1}{2\alpha_1^2}\frac{\partial\lambda}{\partial y} \tag{3}$$

$$w = w_o + \frac{1}{2\alpha_1^2}\frac{\partial\lambda}{\partial z} \tag{4}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{5}$$

Sherman solves the above equations, subject to the boundary conditions

$$n_i \lambda \delta(v_i) = 0 \tag{6}$$

where n_i is the outward positive unit normal in the ith coordinate direction. Correspondingly, $\delta(v_i)$ denotes the first variation of the ith wind component; values of $i = 1,2,3$ correspond to the x,y,z directions. The boundary condition can be satisfied by having either $\delta(v_i)$ or $\lambda=0$. The latter allows adjustment of normal components at the boundary because the corresponding partial derivatives of λ will not necessarily be zero. Sherman (1978) uses $\lambda=0$ to specify boundary conditions along the x and y boundaries. At the upper and lower surfaces, the partial derivative of λ in the normal direction is set equal to zero, as are the initial estimates for the normal component of airflow at those boundaries.

Sherman differentiates Equations (2) through (4) and substitutes Equation (5) to get the following equation that can be solved for λ.

$$\frac{\partial \lambda}{\partial x^2} + \frac{\partial^2 \lambda}{\partial y^2} + \left(\frac{\alpha_1}{\alpha_2}\right)^2 \frac{\partial^2 \lambda}{\partial z^2} = -2\,\alpha_1^2\, \nabla \cdot \vec{V}_o \tag{7}$$

where $\vec{V}_o$ is the initially estimated velocity.

Sherman (1978) used irregular domain boundaries to include terrain features. This requires that the interior points adjacent to cells containing terrain obstacles be treated specially, making it more difficult to base a general application computer program on the original Sherman (1978) approach. The terrain-following coordinate system described in the next section was developed in part to overcome this difficulty.

Bhumralkar et al (1980) changed the coordinate system used by Sherman (1978) in order to simplify the treatment of the lower boundary in their mass-consistent wind interpolation scheme. That work was modified by Endlich et al (1982) to give the version discussed here. In this model, the vertical coordinate (σ) is related to the height (z) in the rectangular coordinate system as follows:

$$\sigma = \frac{z - Z_o(x,y)}{Z(x,y) - Z_o(x,y)} \tag{8}$$

where Z_o is the height of the terrain surface and Z is the height of the top of the domain. It is presumed that the top of the domain conforms to the

streamlines so that there is no upward mass flux from the model volume. The vertical velocity (w) in the rectangular coordinate system is related to that in the sigma system ($\dot{\sigma}$) as follows

$$w = \dot{\sigma} - (\sigma-1)\ \vec{V}\cdot\nabla Z_o + \sigma\vec{V}\cdot\nabla Z \tag{9}$$

Endlich et al (1982) replaced the wind component variables with the following

$$\begin{aligned} u^* &= u(Z - Z_o) \\ v^* &= v(Z - Z_o) \\ w^* &= \dot{\sigma}(Z - Z_o) \end{aligned} \tag{10}$$

Then, if the vertical motions caused by the slope of the sigma surfaces are small compared to the horizontal motions, the continuity equation can be written

$$\frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} + \frac{\partial w^*}{\partial \sigma} = 0 \tag{11}$$

This corresponds to Equation (5), given above for Sherman's (1978) model. Equations (2) through (4) are the same in the Endlich et al. (1982) model as in Sherman's, except that u is replaced by u^*, u_o by u^*_o, and so forth.

Rather than eliminating u^*, v^*, and w^* and solving for λ, Endlich et al solve directly for the wind components using the following equations.

$$\begin{aligned} &\frac{\partial^2 u^*}{\partial x^2} + \frac{\partial^2 u^*}{\partial y^2} + \left(\frac{\alpha_1}{\alpha_2}\right)^2 \frac{\partial^2 u^*}{\partial \sigma^2} \\ &\quad = \frac{\partial^2 u^*_o}{\partial y^2} + \left(\frac{\alpha_1}{\alpha_2}\right)^2 \frac{\partial^2 u^*_o}{\partial \sigma^2} - \frac{\partial}{\partial x}\left(\frac{\partial v^*_o}{\partial y} + \frac{\partial w^*_o}{\partial \sigma}\right) \end{aligned} \tag{12}$$

$$\frac{\partial^2 v^*}{\partial x^2} + \frac{\partial^2 v^*}{\partial y^2} + \left(\frac{\alpha_1}{\alpha_2}\right)^2 \frac{\partial^2 v^*}{\partial \sigma^2}$$

$$= \frac{\partial^2 v_o^*}{\partial x^2} + \left(\frac{\alpha_1}{\alpha_2}\right)^2 \frac{\partial^2 v_o^*}{\partial \sigma^2} - \frac{\partial}{\partial y}\left(\frac{\partial u_o^*}{\partial x} + \frac{\partial w_o^*}{\partial \sigma}\right) \tag{13}$$

$$\frac{\partial^2 w^*}{\partial x^2} + \frac{\partial^2 w^*}{\partial y^2} + \left(\frac{\alpha_1}{\alpha_2}\right)^2 \frac{\partial^2 w^*}{\partial \sigma^2}$$

$$= -\left(\frac{\alpha_1}{\alpha_2}\right)^2 \frac{\partial}{\partial \sigma}\left(\frac{\partial u_o^*}{\partial x} + \frac{\partial v_o^*}{\partial y}\right) + \frac{\partial^2 w_o^*}{\partial x^2} + \frac{\partial^2 w_o^*}{\partial y^2} \tag{14}$$

These equations are obtained by taking appropriate partial derivatives of the basic equations and summing to eliminate v*, w*, and λ to obtain Equation (12). Similar strategies are used to eliminate the appropriate variables and obtain Equations (13) and (14).

Endlich and Lee (1983) assume that there is no flow through the sigma surfaces (analogous to Sherman's use of α_1 and α_2 weighting factors that limit vertical motion to very small values), so the flow in and out of each volume element takes place at the vertical faces. The flux through each face is proportional to the product of the vertical separation Δz between sigma surfaces and the u or v wind component. The divergence for a volume element is given by:

$$D = \frac{\delta u^*}{\delta x} + \frac{\delta v^*}{\delta y} \quad . \tag{15}$$

In the model used here, the vertical component of vorticity is constrained to remain unchanged while the wind components are adjusted to achieve nondivergence. The vertical component of vorticity is given by

$$\psi = \frac{\delta v^*}{\delta x} - \frac{\delta u^*}{\delta y} \quad . \tag{16}$$

Adjustment processes used to achieve local flow nondivergence through the alterations of wind components have been described by several authors (e.g., Endlich, 1967; Liu and Goodin, 1976; Chen, 1980; Goodin et al, 1980). Endlich's procedure begins by decreasing (or increasing) the wind components on the inflow faces and increasing (or decreasing) those on the outflow faces in order to correct for a net convergence (or divergence). The wind component changes required to produce nondivergence in a volume element are usually

quite small. Further changes are made to neighboring wind components to return the vorticity to the value originally estimated for the element. The volume elements are adjusted iteratively until the divergence (either positive or negative) is reduced to very small magnitudes while retaining the original vorticity fields. Typically, 10 to 15 complete iterations are required. This technique allows the model given in the appendix to make the flow nondivergent in each layer independently of the flow in neighboring layers, which requires less computation time than the full 3-dimensional relaxation process. The subroutine BAL5 performs the iterative adjustment described above.

The nondivergent wind model is run for the mean wind components at each observation site (and the mean geostrophic components aloft), and for the input data sets represented by each of the eigenvectors. The resulting winds at the top and bottom of the layer are then stored along with the corresponding input data sets (the mean winds, the eigenvector wind sets, and the elevations of the top and bottom of the boundary layer) for use by the transport and diffusion model. It should be noted that the Endlich and Lee model uses cm s^{-1} units for wind outputs and cm units for elevations. The diffusion model converts to m s^{-1} and m units. If other wind model results are used as inputs, some code modification may be required.

C. Transport and Diffusion Model

One of the simplest approaches to the modeling of a plume is to represent that plume by a series of overlapping "puffs." This approach can accommodate winds that vary in space and time, but it can also have large computational and computer memory requirements if it is applied indiscriminately. The implementation of this concept used here assumes a Gaussian distribution of concentrations within each puff; that distribution is characterized by a standard deviation σ_z in the vertical and a radially symmetric horizontal standard deviation, σ_y. In the limit, when many puffs each containing a small amount of material are used, the behavior of a continuous Gaussian plume can be reproduced quite accurately. The number of puffs required to simulate the plume depends on their size; larger puffs can be farther apart. Thus, fewer puffs will be needed to simulate the plume accurately as the puffs grow.

1. Reducing the Required Number of Puffs

Ludwig, Gasiorek, and Ruff (1977) determined that the minimum spacing between puffs should be $2\sigma_y$ or less. With such spacing, the concentrations derived from a series of puffs will be within about five percent of those from a continuous Gaussian plume. This information has been used in two ways to reduce the number of puffs that are required. First, closely spaced puffs are merged periodically so that one puff containing all of the material in the original two puffs replaces the original two. The other parameters (puff location, σ_y and σ_z, age, and so forth) are averaged.

The second approach to reducing the required number of puffs is based on the fact that although it is common to discuss "point" sources, sources

generally have finite dimensions. Even if they did not, the plume would quickly expand so that it is reasonable to use puffs whose initial dimensions are finite. This may prevent accurate simulation of plume behavior in the immediate vicinity of the source, but it allows puffs to be generated at a rate that is inversely proportional to wind speed. A newly created puff moves away from the source until it reaches a distance of about 2 σ_y before the next puff is released. If puffs were released at a constant rate, that rate would have to be set so that it could accommodate the highest expected wind speed. This would lead to many more puffs than are usually necessary. For the model described here, a new group of puffs is generated every ten minutes. The rate at which they are generated is such that they will be within σ_y of one another after about 5 minutes travel time.

The amount of material within each puff is determined by two factors, the emission rate and the rate at which puffs are generated. Upon creation, each puff is assigned an amount of material Q equal to the emission rate divided by the frequency of puff release.

Puffs are eliminated from consideration once they pass outside a square that is used to define the computational domain. If the source is located near the center of this square, puffs can travel more than 15 km from the source before they are eliminated. The computational domain can be expanded by changing the grid spacing (set at 6 km in the current code) or the numbers of grid squares (a 10 x 10 array is used). However, any change in the computational grid will require a corresponding change in the wind field inputs.

2. Puff Advection

The location of each puff center must be known for the concentration calculations to be made. The model calculates the position of the puff centers at 2-minute intervals. Temporal changes in the wind field are allowed to occur every ten minutes. During a ten-minute interval between wind changes, the wind field is assumed to remain constant. The motion of a puff during a 2-minute interval is determined by the wind in the grid square where the puff center is located at the beginning of the period. The puff center is moved in a straight line parallel to that wind direction for a distance equal to a distance that the wind would carry it in 2 minutes.

3. The Form and Growth of the Puffs

As noted before, each puff is considered to be horizontally symmetric with a Gaussian, concentration function of the following form (cf Slade, 1968):

$$C = \frac{2Q}{(2\pi)^{3/2}\,\sigma_y^2\,\sigma_z} \exp\left[-\frac{1}{2}\left(\frac{y}{\sigma_y}\right)^2\right] \exp\left[-\frac{1}{2}\left(\frac{z}{\sigma_z}\right)^2\right] , \qquad (17)$$

where

C = concentration at ground level (g m^{-3})

Q = total material in the puff (g)

σ_y, σ_z = standard deviations of the Gaussian distributions in the horizontal and vertical directions, respectively (m)

y = horizontal (radial) distance from the puff center (m)

z = height of the puff center above the ground (m).

The terms, σ_y and σ_z define the "size" of the puff and depend on the distance traveled by the puff and the prevailing atmospheric stability. If the atmospheric stability has remained constant throughout its travel, then they are approximated by functions of the following form:

$$\sigma = a_j x^{\gamma_j} \tag{18}$$

where a_j and γ_j are constants depending on the stability class j, and x is the distance traveled by the puff. The values of the constants used by the model are given in Table 1. Modeling the effects of changes in atmospheric stability requires some modification of the basic equation through the use of a virtual travel distance, x_v (Ludwig, 1982). This is the distance that satisfies Equation (18) for the current value of σ. If there has been no change in stability during the growth of the puff, x_v will equal x. Rewritten, Equation (18) becomes

$$\sigma_\alpha = a_{\alpha j} \left(x_{\alpha v}\right)^{\gamma_{\alpha j}} \tag{19}$$

Equation (19) can be used to calculate σ_y or σ_z if we recognize that there will be discontinuous changes in $x_{\alpha v}$ whenever the atmospheric stability changes. During periods when the atmospheric stability remains the same, the increase of $x_{\alpha v}$ is the same as the increase in travel distance, i.e., for a period when stability remains constant,

$$\left(x_{\alpha v}\right)_{i+1} = \left(x_{\alpha v}\right)_i + \Delta x_i \quad , \tag{20}$$

Table 1

CONSTANTS USED TO DETERMINE σ_y AND σ_z FROM EQUATION (18)

Atmospheric Stability Class*, j	σ_y		σ_z	
	a_{yj}	γ_{yj}	a_{zj}	γ_{zj}
1	0.428	0.9	0.025	1.38
2	0.301	0.9	0.075	1.07
3	0.198	0.9	0.112	0.91
4	0.128	0.9	0.196	0.72
5	0.095	0.9	0.231	0.64
6	0.065	0.9	0.227	0.575

* 1 = Extremely unstable
2 = Moderately unstable
3 = Slightly unstable
4 = Neutral
5 = Slightly stable
6 = Moderately stable

where $(x_{\alpha v})_i$ and $(x_{\alpha v})_{i+1}$ are the values of $x_{\alpha v}$ at the beginning and end of the period and Δx_i is the actual distance travelled during the period. If the stability changes at the beginning of the period, then

$$\left(x_v\right)_\alpha = \left(\frac{\sigma_\alpha}{a_{\alpha j}}\right)^{1/\gamma_{\alpha j}} \tag{21}$$

where the stability during the new period is in class j, and $a_{\alpha j}$ and $\gamma_{\alpha j}$ are the appropriate constants for Equation (19). It should be noted that the virtual travel for σ_z will generally be different from that for σ_y. The values of Δx are calculated in the part of the program that simulates the advection of the puffs. Changes in atmospheric stability are restricted to occur only once per hour.

Often there are restrictions to the vertical expansion of the puff. Elevated temperature inversions can cap the "mixing layer" and effectively prevent the further upward expansion of a puff. Such behavior is commonly treated (e.g., Turner, 1970) by assuming that puff behavior on encountering the top of the mixing layer is analogous to reflection. Thus, the concentration at a monitor is equal to that part directly reaching the ground, as in Equation (17) and another part "reflected" from the top of the mixing layer. The reflected contribution is simulated by invoking a "virtual" source above the reflecting surface. If the height above ground of the mixing layer is H

and the height of the real puff is z, then the height, z', of the virtual puff is given by

$$z' = z + 2(H-z) = 2H - z \quad . \tag{22}$$

Later, when the calculation of concentration at a receptor is discussed, we will return to the virtual puff.

As the size of the puff increases, reflections from the ground and the top of the mixing layer cause the distribution of concentration in the vertical to become nearly uniform between these two surfaces. In such a case, the concentration is proportional to the total material and inversely proportional to the depth of the mixed layer. Concentration is then given by

$$C = \frac{Q}{2\pi H \sigma_y^2} \exp\left[-\tfrac{1}{2}\left(\frac{y}{\sigma_y}\right)^2\right] \quad . \tag{23}$$

As long as the mixing height remains constant, or increases with time, application of the equation with the current value of the mixing height presents no problem. However, if mixing height is decreasing with time, the equation would indicate that concentration was increasing with time. This would be contrary to nature, because the upward or downward movement of the mixing height does not involve motions of a physically impervious surface; i.e. it does not act like a piston recompressing concentrations. The model assumes that once the puff has expanded to the point where it is affected by the top of the mixing layer, subsequent reductions in the height of that layer will not increase concentrations. Thus, once the top of the mixing layer is encountered, the height through which the puff is mixed is only allowed to increase not decrease. The model keeps track of the greatest mixing height that has been encountered during the history of the puff and uses that value to estimate reflection effects.

4. Concentration Calculations at Receptors

The model described here is designed to calculate concentrations only at receptors near the surface. The user must select the receptor spacing and then the model defines a square grid of receptors centered at the source location. The receptor array is a 21 x 21 grid. In order to keep the receptor array within the computational domain, grid spacing should generally be less than 3 km. For smaller grid spacing, puffs may pass outside the receptor grid without being discarded. This allows for recirculation of puffs over receptors as long as the puff centers did not pass outside the 60 km square computational domain.

It would require very many calculations to consider the contribution of each puff to each receptor. Therefore, if a puff center is more than 3 σ_z above the surface, or more than 3 σ_y horizontally from a receptor, its contribution at that receptor is considered to be nil. In order to accomplish this, the entire list of puffs is scanned when the concentrations are to be calculated. The receptors that are within 3 σ_y of a puff center are determined and the contribution to the concentration at these receptors is calculated and added to the contributions of other nearby puffs. Concentrations are calculated once per hour.

III STRUCTURE OF THE COMPUTER CODE

A. General Organization

Figure 2 is a flow chart showing the organization of the MADICT computer code. As the figure shows, the program begins by calculating some of the values that will be used repeatedly. The locations of the receptors are defined and those quantities that are presumed not to change during the course of a run are input; these include the location of the source and the eigenvectors appropriate to the set of wind stations that are being used. Next, the time varying meteorological conditions are input. These are changed hourly; the program can be stopped by entering a negative mixing depth. The program interpolates between winds at the beginning and end of the hour (other meteorological factors such as mixing height and stability are assumed to remain constant through the hour), so that it is necessary to enter two sets of winds at the beginning of a run. For later hours, the winds at the end of the upcoming hour are entered (the winds for the end of the preceding hour are used as the initial winds for the upcoming hour). During the course of the hour the winds are interpolated using the eigenvector coefficients for the beginning and end times; winds are assumed to remain constant over 10-min intervals.

After the meteorological data have been input, the program determines whether or not the appropriate wind field solutions (for the current mixing depth) have been used during the preceding hour (if it is not the first hour). If not, the appropriate solutions are read from the disk. Interpolated eigenvector coefficients are then calculated, and the fact that no winds have been determined for any grid cell is indicated by making the vertical component in each wind cell equal to the x coordinate of the origin. It should be noted that the calculation grid is different from the receptor grid. The receptor grid is centered on the source and has a spacing that is specified as an input. The calculational grid is a geographical grid where distances are expressed in meters. It defines the computational domain which is 30 km on a side and has its origin at (-15,000, -15,000, 0). Winds are determined on a 10 x 10 grid at the top and at the bottom of the domain. All trajectories are calculated in terms of this computational domain.

Old puffs are advected every 10 minutes (in 2-min time steps) and new puffs are generated and moved to their appropriate initial positions according to the wind at the source. During the first of these 10-minute time periods for each hour, the stability is checked to see if it has changed from the previous hour (or if it is the first hour of the calculation), in which case new virtual travel distances corresponding to the current puff dimensions are calculated. At the end of each 10-minute time period, the list of puffs is examined and those which are overlapping sufficiently are merged. Those which are outside the computational domain are discarded from further consideration.

The receptor concentrations are accumulated after the six 10-minute time periods in the hour have been cycled through. The puffs are treated sequentially, first determining which receptors they influence and then adding their

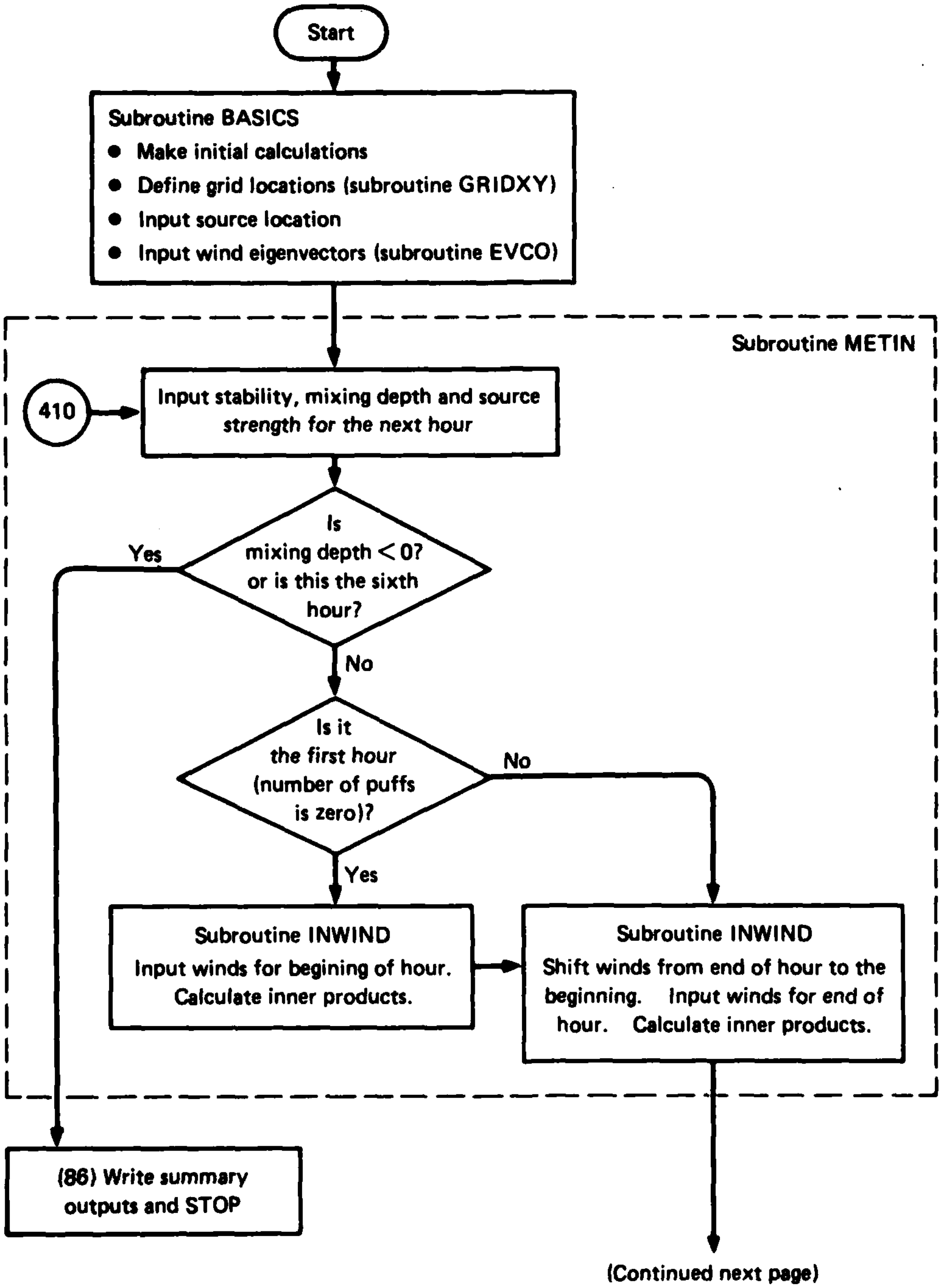
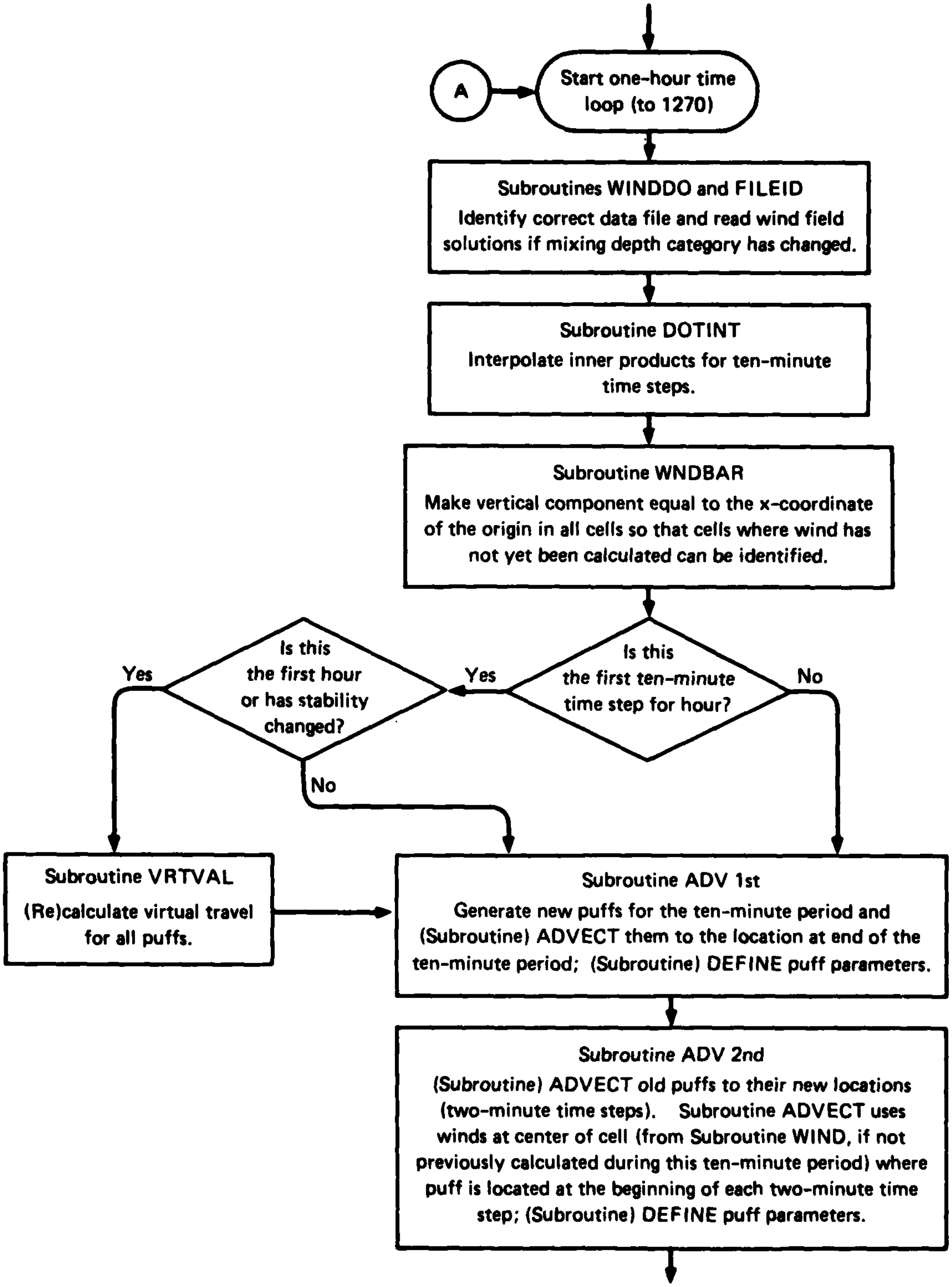


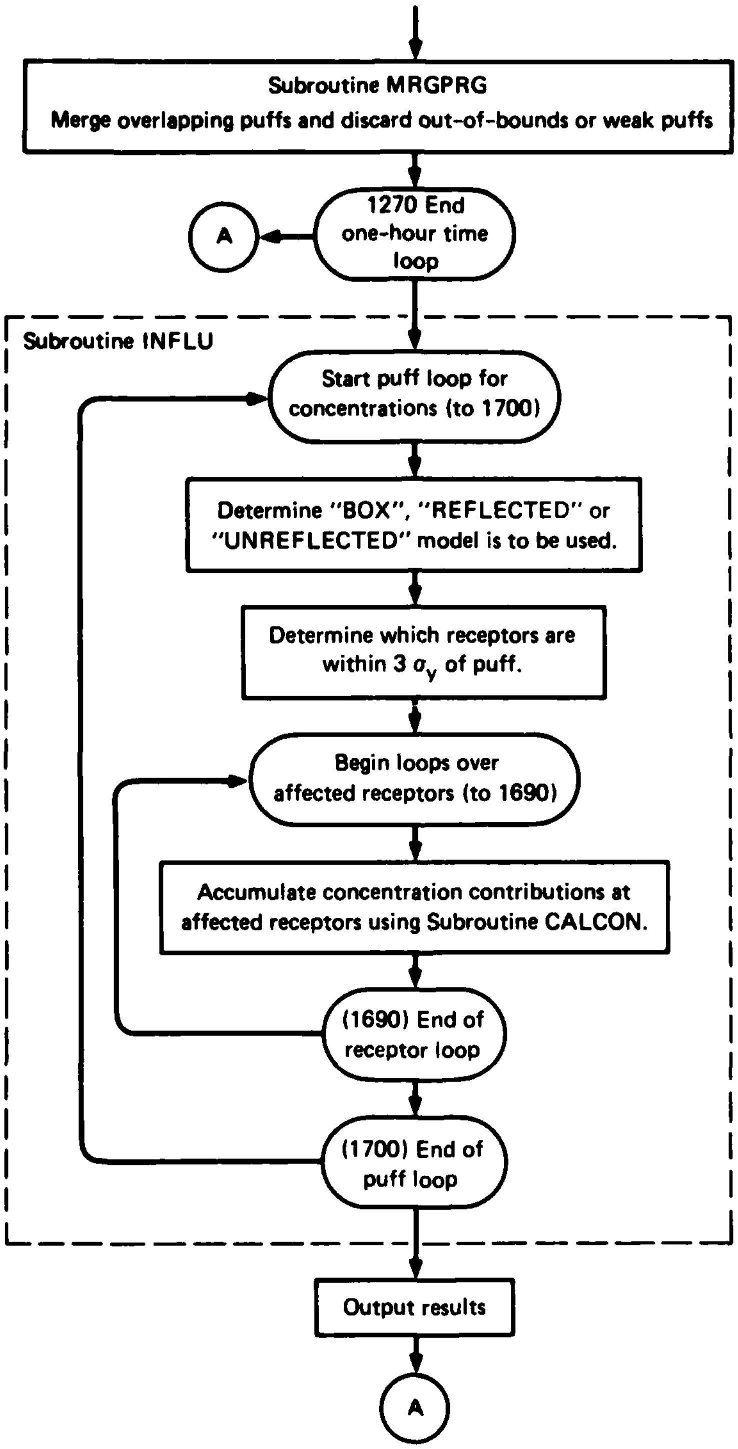
FIGURE 2 FLOW CHART FOR MADICT

(Continued from previous page)



(Continued next page)

(Continued from previous page)



concentration contribution to the concentrations of earlier puffs at those receptors. After all puffs have been treated, the results are output. The program includes a subroutine to be used for calling a local graphics package. A National Center for Atmospheric Research (NCAR) graphics package routine is used as an example. The concentrations for the hour at each receptor are also stored. After the program has run for five hours, or has been stopped by the input of a negative mixing height, all non-zero receptor concentrations for the simulated time period are printed.

B. Major Components

This section provides brief descriptions of the various subroutines and functions. A listing of the FORTRAN77 code is given in Appendix B.

1. Subroutine ADVECT (XX, YY, ZZ, TRAVEL)

This subroutine advects the puff from the initial coordinates (XX, YY, ZZ) to determine the location at the end of five 2-minute steps. This new location is returned through the same arguments (XX, YY, ZZ). The wind at the beginning of each step is determined by a call to the subroutine WIND. The local wind component (in meters/10-min) are multiplied by the time step to give the new coordinates. Altitude (ZZ) is not allowed to be negative or above the top of the domain. Total travel distance along the trajectory (travel) is also calculated.

2. Subroutine ADV 1ST (LASTPF, VS, VDISRO, VDISZO, ISTAB)

This subroutine generates new puffs at 10-min intervals. It first determines the wind components at the source, and the number of puffs to be generated is calculated so that their spacing will be approximately equal to the value of σ_y (or 20 m, whichever is greater) after 3 minutes of travel. The appropriate initial values of virtual travel (VDISRO, VDISZO) for the stability class (ISTAB), the total material in the puff and so forth are assigned either in this subroutine or through a call to a subroutine DEFINE. The argument LASTPF is the number of puffs at the end of the 10-minute step.

3. Subroutine ADV 2ND

This subroutine uses the subroutine ADVECT to move the puffs that exist at the beginning of the 10-min time period to their new locations. The puff parameters are then updated with a call to the subroutine DEFINE. The subroutine ADVECT determines when puffs are out of bounds and sets a logical flag (LOUTFL) equal to .TRUE., which in turn causes ADV 2ND to set the amount of material in the puff to zero. This causes the puff to be purged from the list.

4. Subroutine BASICS (VDISRO, VDISZO, IHOUR)

This subroutine calculates some basic parameters such as the initial virtual travel distances (VDISRO, VDISZO) for the six stability classes. It also calculates and stores values of exponent EXP (-0.1*I) in the array FEXP(I). It also interactively requests information for defining the receptor grid, the characteristics of the wind inputs, source location and the beginning time. It calls the subroutine EVCO which reads the empirical orthogonal functions from the appropriate file.

5. Subroutine CALCON (ITYPE, ISTAB, ZZ, TYPMDL, QMULTF, II, JJ, I, SIGZ, NHOUR)

This subroutine accumulates concentrations for a given puff/receptor pair. The puff is defined by I and the receptor by II, JJ. The value of the argument QMULTF takes into account the amount of material in the puff and the horizontal separation between the puff center and the receptor. This subroutine uses one of three (determined by ITYPE) models of the vertical distribution--a box model (ITYP = 1), or a model where the vertical expansion is not affected by the elevated mixing layers (ITYP = 2), or finally (ITYP = 3) the case where the puff is "reflected" from the top of the mixing layer. TYPMDL is used to redefine virtual travel to be consistent with whatever restrictions are imposed by the mixing height on vertical growth. NHOUR defines the time for which concentrations are being calculated.

6. Subroutine DEFINE (IPUFF, XX, YY, ZZ, TRAVEL)

This subroutine is used to update the parameters associated with each puff. These parameters are stored in two arrays, PUFINT (IPUFF, J) and PFFPAR (IPUFF, K). In both arrays, the puff is identified by the first index. the second index (J) for PUFINT refers to maximum mixing depth (m) encountered, the three spatial coordinates (m) and the time (min) since the puff was released for values of J from 0 to 4, respectively. The second index (K) for PFFPAR refers to the amount of material in the puff (gm) for (K = 0); the indices K = 1,2 refer to the virtual travel distance for σ_z and σ_y (m), respectively.

7. Subroutine DOTINT (ITIME)

This subroutine interpolates the inner products at the beginning and end of the hour so that they are appropriate to the time step, ITIME. The current interpolated coefficients of the eigenvectors are stored in the array DOTP.

8. Subroutine EVCO

This subroutine reads the eigenvectors and the mean winds. (Units are corrected to m s^{-1} from the cm s^{-1} that are output by the wind field

preprocessor programs; use of other preprocessors may require some modification of the MADICT code). The files created by the Endlich wind programs identify the means as the zeroth eigenvector; furthermore, the first eigenvector is the one which explains the least variance, the second explains the second least variance and so forth. This subroutine stores the means separately as UM and VM. This subroutine also reverses the indexing on the eigenvectors so that the lowest index refers to the eigenvector that accounts for the largest amount of the variance, the second for the second most variance.

9. Subroutine FILEID (IREAD DEPTH)

This subroutine reads the first record from each of the wind field source files. The first record in each of these files includes information about the average mixing depth that was assumed in calculating the solutions. The subroutine identifies the solutions with the assumed mixing depth that is the closest to the currently observed mixing depth (DEPTH). The appropriate logical unit (IREAD) is returned so that the correct wind field solutions are read and stored. The PARAMETER NFILES in this subroutine must be set to the number of mixing depth categories for which solutions have been stored.

10. Subroutine GETWND (IX, IY, IZ)

The subroutine calculates the wind in the cell IX,IY,IZ from the stored solutions. It does this by summing the u, v, w components for each eigenvector solution, weighted according to the inner product of the observations and that particular eigenvector. Winds are calculated at the top and the bottom of the layer for the solutions that have been stored for those two elevations. Winds at intermediate altitudes are obtained by log-linear interpolation to the centers of the intermediate grid cells. The units of the stored winds are meters per 10-minute interval. The subroutine assumes that the input wind field solutions are in cm s^{-1} units, as provided by the wind field preprocessors discussed in this report. For wind field solutions from other sources, this subroutine may require modification.

11. Subroutine GRAFF (I,DBUG)

This subroutine was included to provide a mechanism for users to introduce graphical presentations. It uses an NCAR graphics routine.

12. Subroutine GRIDXY (DBUG)

This subroutine calculates the x- and y-coordinates of the grid points in the receptor grid. The receptor grid is centered on the source and the coordinates are given in terms of the coordinate system used to describe the modeling domain.

13. Subroutine INDXY (XX, YY, LX, LY, LOUTFL)

This subroutine returns the indices (LX and LY) of the grid cell in which the coordinates XX and YY fall. It also sets the logical variable LOUTFL to .TRUE. when the coordinates are outside the domain; otherwise it is .FALSE..

14. Subroutine INFLU (GSIGY, NHOUR, ISTAB)

The purpose of the subroutine is to identify those receptors which are significantly influenced by each puff (i.e. those receptors within 3 σ_y). It uses the subroutine CALCON to increment the concentrations at those receptors that are influenced. The subroutine also determines whether or not concentrations are uniformly distributed in the vertical (box model), reflected from the upper domain boundary (the reflected model) or still freely expanding in the vertical (the unreflected model). The effects of the amount of material in the puff and the concentration distribution in the vertical are calculated before the subroutine CALCON is called.

15. Subroutine INWIND (I, IHOUR)

This subroutine is used to input the observed winds, from which the wind components are calculated. After the wind components have been calculated, the mean components are subtracted and the inner products, DOTX (I, J), are calculated for the beginning (I=0) and end (I=1) of the hour for each eigenvector (indicated by the second index).

16. Subroutine MET IN (QUIT, ISTAB, IHOUR, ALPHAY, SEPMAX)

This subroutine reads the basic meteorological information for the upcoming hour. The logical flag QUIT is set to .TRUE. if a negative mixing depth is input; this will cause the program to stop. Source strength (QSOURC) is input in gs^{-1}. This subroutine calls the subroutine INWIND to get the observed wind inputs.

17. Subroutine MRG PRG (LASTPF, SEPMAX)

This subroutine scans the list of puffs and determines those which are close enough together to be merged. Average parameter values are calculated for each pair to be merged, and those values are assigned to one of the puffs along with the total material from the two puffs. The other puff is assigned zero material. After the list has been scanned, those puffs with zero material are discarded and the remaining puffs are renumbered accordingly. Those puffs which have passed outside the modeling domain are also removed because such puffs have their content set to zero in the subroutine ADV 2ND.

18. Function NEXTHR (IHOUR)

This simple function increments the hour and ensures that the time remains between 0000 and 2300.

19. Subroutine OUTNOW (NHOUR, ISTAB)

This subroutine prints the calculated concentrations for all the hours considered and the most recent meteorological parameters when all the calculations are completed. Only non-zero concentrations are printed.

20. Subroutine RELHT (RHS, XRHS)

The array of grid cell terrain elevations and mixing heights XRHS is scanned to determine the lowest point at the surface. All other elevations (at the surface and the top of the domain) are transformed so that they are measured relative to the low point. The transformed heights are returned in the array RHS.

21. Subroutine VIRTVL, (ISTAB, SL)

This subroutine is called whenever there is a change of stability class from one hour to the next. First, the current values of σ_y and σ_z are determined from the virtual travel distances and the stability class (SL) for the preceding hour. The virtual travel distances are recalculated for the new stability class (ISTAB).

22. Subroutine WIND (KEY, XX, YY, ZZ, U1, U2, U3)

This subroutine is used to determine if a puff is outside the modeling domain. If so, the flag LOUTFL is set to .TRUE. so that the puff will be discarded later. If a wind has not been calculated for the cell, the subroutine GETWND is called. Otherwise, the u,v, and w components (U1,U2,U3) that have previously been stored in the array WNDUVW(IX,IY,IZ,I) are returned; the index I denotes the component (I=0 for u to I=2 for w).

23. Function WINDIR (WD)

This function converts the wind direction in meteorological convention (i.e. the direction from which the wind blows--measured clockwise from north) to mathematical vector convention (i.e. the direction toward which a vector points, measured counterclockwise from east). The wind direction is also converted from degrees to radians.

24. Subroutine WINDDO

This subroutine reads the appropriate precalculated windfields from the logical unit IREAD. The appropriate logical unit is determined by a call to subroutine FILEID. The precalculated solutions are stored in the array PRECAL (IX,IY,LEVEL,ICOMP,IGEN). The first two indices refer to the grid cells in the x and y directions numbered from 0 to 9. The index, LEVEL, differentiates the top and the bottom of the domain; 0 denotes the bottom and 1 the top. ICOMPS designates the wind components (0=u, 1=v, 2=w). The final argument, IGEN, identifies the solution, with a value of 0 for the mean wind solution, 1 for the eigenvector explaining the most variance and so forth. This subroutine stores the ID of the logical unit from which solutions were last read (LAST) so that a new set of solutions will be read only when there has been a significant change in the mixing depth.

25. Subroutine WINDBAR (WNDUVW, YORIGON)

This subroutine sets the w components stored at the beginning of a 10-minute period in the array WNDUVW equal to the y coordinate of the grid origin. As noted before, this allows identification of grid cells for which winds have not yet been calculated.

IV USE OF THE MODEL

A. Preprocessing of The Wind Information

1. General

As noted earlier, the preprocessing of the data makes use of programs that have been developed and described in detail by Endlich and Lee (1983) and Endlich et al (1980). Appendix A provides a listing of preprocessing programs. The remainder of this section provides a brief overview of how those programs are to be used. If more details are needed, the original references should be consulted. Briefly, the procedure requires that an extended period of wind data be available (up to one year) so that those data can be used to derive the necessary eigenvectors of their covariance matrix. These results are then input with terrain information into the program which calculates mass-consistent flow patterns. These mass-consistent flow patterns will be developed for several mixing height categories. Results of the computations are then stored in the form that can be easily used by the diffusion model. A block diagram showing the steps required to derive wind statistics and the computer programs used is given in Figure 3.

2. Programs in the Preprocessing Package

(a) Program TERRAIN

This program reads digital terrain data obtained on magnetic tape (nine track, 1600 bpi) from the National Cartographic Information Center, Reston, Virginia 22092 (telephone 703/860-6045). Each tape contains data for up to seven 1° lat.-long. regions These data are read from logical unit 1. The program selects part of the original dense data and averages it to produce smoothed height values for small areas having sides of 0.01° in lat. and long. The output is an array HT(101,101) of smoothed height values with point (1,1) at the southwest corner. The array is written in a file on logical unit 2. We will refer to this information as output A0.

The program also gives a printout of the smoothed height data. The height interval of the symbols is controlled by the parameter HTINT, which is read from logical unit 5 in the format F6.1. The listing of this program is given in Appendix A.

(b) Program GRIDAT

This program selects grid point values of terrain height from tapes of type A0 (see Program TERRAIN). The grid point values of terrain height are assembled in the array GHT(22,22). GHT(1,1) is at the southwest corner of the array. The input parameters are:

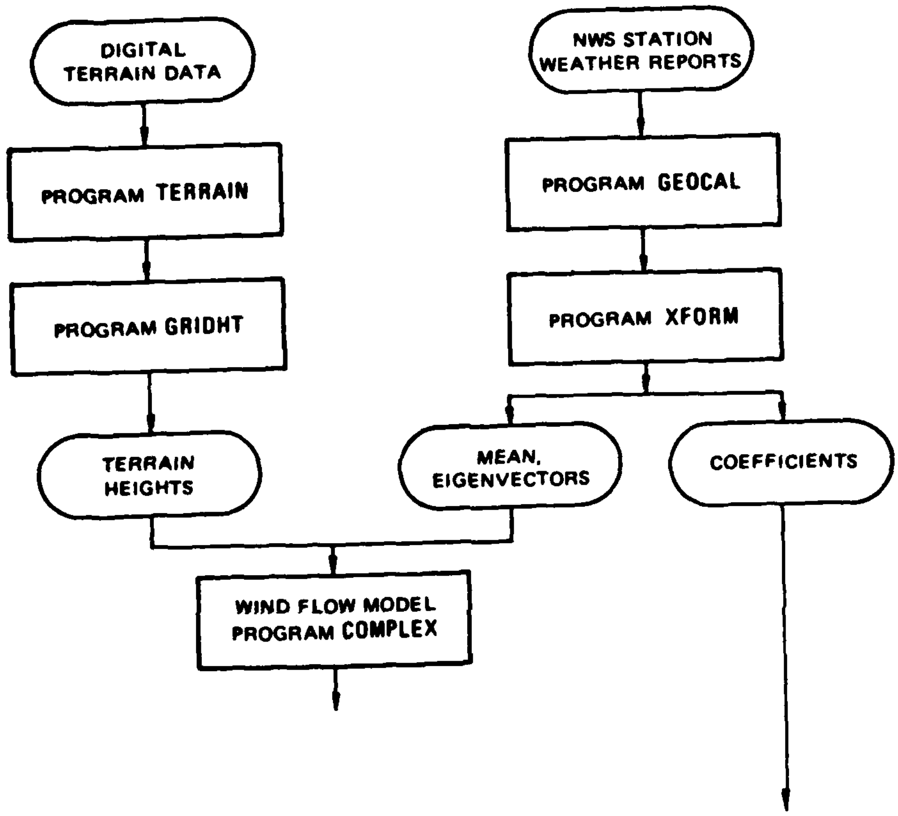


FIGURE 3 BLOCK DIAGRAM OF WIND PROCESSOR PROGRAMS (AFTER ENDLICH AND LEE, 1983)

MX -- Number of columns in the terrain height array

NY -- Number of rows in the terrain height array

NGCX,NGCY -- Column and row of the site

GINCX,GINCY -- Grid increments (km) in the x and y directions

NFILES -- Number of files of data on the input tape

SLAT,SLNG -- Latitude and longitude of the site in degrees and hundredths

SHGT -- Actual height of the site

NFILES -- Number of files of data on the input tape.

Table 2 gives the formats of the input parameters.

Table 2

INPUT FOR PROGRAM GRIDHT

Input Record No.	Variables	Format
1	MX, NY, NGCX, NGCY GINCX, GINCY	4I5,2F6.1
2	SLAT, SLONG, SHGT	3F10.3
3	NFILS	I5

The array GHT is written (or punched on cards in the version given here) in the format 11F6.0 for use by the windflow model (COMPLEX). The listing of this program is given in Appendix A.

c. Program GEOCAL

This program creates a file of weather data for groups of weather stations. The file includes u and v components (m s^{-1}) and geostrophic wind computed from sea-level pressure data. A stability index is also computed from cloud and temperature data. The file is written on logical unit 2.

The input data are National Weather Service TD-1440 Airways Surface Observations, Card Deck 144, on nine-track, 1600 bpi, EBCDIC magnetic tapes. They are read on logical unit 1. Each tape contains a year of data for 4 to 6

stations. The weather data may be obtained from the National Climatic Data Center, Federal Building, Asheville, North Carolina 28801 (telephone 704/258-2850). The program fills in missing or garbled data with the last available reliable observation.

The input parameters (read from logical unit 5) are:

NSTA -- Number of weather stations used

IU1, IU2, IU3 -- Indices of stations used in geostrophic wind computation

IDATES -- Starting date in terms of year, month, day (example 770101)

IGHS -- Starting hour (GMT)

ISTA -- Weather station identification numbers used by NCDC

IGMT -- Time corrections to convert local time to GMT for weather stations

ALAT -- Latitude of weather stations

ALONG -- Longitude of weather stations

IFMT2 -- Header format for data printout (on logical unit 6)

IFMT5 -- Data format for printout.

Table 3 gives the formats for the input cards.

Table 3

INPUT FOR PROGRAM GEOCAL

Input Record No.	Variables	Format
1	NSTA, IU1, IU2, IU3 IDATES, IGHS	10I8
2	ISTA(L)	10I8
3	IGMT(L)	10I8
4	ALAT(L)	10F8.2
5	ALON(L)	10F8.2
6	IFMT2	8A10
7	IFMT5	8A10

The output is a file on logical unit 2 of wind components and stability at each station, and a geostrophic wind for the area. The data are in synoptic order. For 1977, the data are at three-hourly intervals. This output is referred to as CO.

The listing of this program is given in Appendix A.

d. Program XFORM

This program makes use of subroutines from the International Mathematical and Statistical Library (IMSL). A detailed description of its use in wind energy evaluation was given by Bhumralkar et al (1978). The program calls IMSL subroutines to compute a covariance matrix of the input data, the eigenvectors of the covariance matrix, and the coefficients of the eigenvectors. Other subroutines could be substituted to perform these functions. The calling parameters are described in detail in the listing of this program given in Appendix A.

The input data are station wind data in terms of u and v components and geostrophic wind components at three-hourly intervals from tapes of type CO (see program GEOCAL).

The output includes a listing of the mean winds, the eigenvectors, and the percentage of variance explained by each eigenvector. The means and eigenvectors are punched on cards (in the version given here) for use by the windflow model. Also, the coefficients of each eigenvector at each time are written in a file on tapes designated DO. These coefficients are not used in the application with the transport and diffusion model.

e. Program CMPLX3

This program comprises the windflow model, which is a modified version of the COMPLEX model described by Endlich and Lee (1983). The model computes nondivergent winds that conform to the terrain and to the shape of the boundary-layer top. The program requires terrain heights for a coarse grid and a fine grid as given by Program GRIDHT. It also requires the mean winds and the significant eigenvectors from Program XFORM. The input parameters are:

JSITE -- Site identification number

NWIND -- Number of wind patterns (data sets) to be treated

NGRID -- Number of grids to be used

IXZ,JYZ -- Column, row of the site in the coarse grid

IXSS, JYSS -- Column, row of the site in the fine grid

HSITE -- Elevation of the site in feet (not used for this application)

MI, N1 -- Number of columns, rows in the coarse grid

MR, NR -- Number of columns, rows in the fine grid

IZ -- Ratio of coarse grid spacing to fine grid spacing in the x direction

JZ -- Ratio of coarse grid spacing to fine grid spacing in the y direction

DSI -- Coarse grid increment (km)

DSR -- Fine grid increment (km)

AVTHK -- Average thickness of the boundary layer (m)

SLFAC -- Slope factor for the boundary-layer top (see Endlich and Lee, 1983)

STHK -- Minimum boundary layer thickness over high terrain (m)

NREL -- Upper limit on the number of relaxations permitted (see Section II-B)

RATIO -- Ratio of vertical to horizontal wind alterations

IPNCH -- Punch control ($\geq$ 20 punches output); program should probably be changed to write records to disk.

Additional parameters used in output are:

IV -- Eigenvector number

UV -- u component of site wind (not used for this application)

VA -- v component of site wind (not used for this application).

The input record formats are shown in Table 4.

The output of CMPLX3 is a field of u, v, and w wind components corresponding to the input wind pattern.

The principal subroutines are:

TOPO -- Reads in the terrain heights and computes the relative heights of the sigma surfaces at each mesh point

SETBLT -- Computes the height of the boundary-layer top based on the input values of thickness and slope

INWND -- Computes the initial estimate of winds for COMPLEX based on the station winds at anemometer height for a wind pattern (i.e, the mean winds or an eigenvector) and the associated geostrophic wind at the upper boundary. Values of wind of intermediate heights are determined by logarithmic interpolation. The subroutine also interpolates initial winds for the fine grid from the altered winds of the coarse grid

Table 4

INPUT FOR PROGRAM CMPLX3

Record No.	Variables	Format
1	JSITE	1I5
2	NWIND, NGRID	2I5
3	IXZ, JYZ, IXSS, JYSS, HSITE	4I5, F10.2
4	MI, NI, MR, NR, IZ, JZ, DSI, DSR	6I5, 2F10.2
5	AVTHK, SLFAC, STHK, DNI	F10.1, F10.2, F10.1, I15
6	NREL, Ratio, IPNCH	I5, E10.1, I5
Output Record		
1	JSITE, IV, UA, VA, DNI, AVTHK, SLFAC, STHK	2I5, 2F10.2, I5, F10.0, F5.1, F8.0

NET -- Interpolates grid-point values of wind components at anemometer height from the station values for each wind pattern being processed

RELAX3 -- Alters the initial winds to a nondivergent condition.

B. Running MADICT

1. General

The program MADICT can be run interactively; some fixed input files are required for the eigenvectors and the wind field solutions. Those file requirements are described below. Examples illustrating the interactive inputs are presented in Appendix C.

2. File Requirements for Preprocessed Wind Information

(a) Eigenvectors and Mean Winds

The eigenvectors and mean winds are read from logical unit IVREAD, (assigned the value 11 in the subroutine EVCO). The format of the file is given in Table 5. The program assumes the values that follow when IV = 0 are the mean components (UM and VM) for each of the NWINDS stations; NWINDS

includes all the ground level sites plus one upper level wind used in defining the eigenvectors. The code also assumes that N = NEIGN+1 is the eigenvector that explains the most variance, and lower IV values are associated with successively less important eigenvectors. NOTE: NEIGN=9 is used in PARAMETER statements in various subroutines. An appropriate value should be substituted for cases where the total number of eigenvectors is not 10 (equivalent to 5 wind observations in an input data set.

Table 5

EIGENVECTOR INPUT FILE
(Logical Unit = 11)

Parameter	Format	Remarks
IV	I10	Identifies Eigenvector Number (see text)
UE, VE or UM, VM	2F10.2	u and v components of eigenvector or mean winds.

Finally, it is best to arrange the eigenvector input data file with the means first, followed by the eigenvectors in order of decreasing importance. Note also that the magnitude of the input eigenvectors should be 100 cm s^{-1}. An example of an input eigenvector file is given in Table 6. The example given would not permit the use of more than four eigenvectors for reconstructing wind field solutions.

(b) Wind Field Solution Sets

The solution wind fields are read from logical units 21, 22 ... (20 + NFILES) where NFILES is the number of solution sets stored. Each file of solution sets should correspond to a different mixing depth category. The number of such files (NFILES) is currently set to 2 in a PARAMETER statement in the subroutine FILEID.

The formats on these files should be as shown in Table 7. The files for logical units, 21, 22 etc. should be arranged so that they correspond to solution sets for increasing mixing depth. Within each file, the first portion should be the solution for the mean, with subsequent sections giving solutions for eigenvectors that account for decreasing amounts of the wind variance. Examples of these files are given with the sample problems in Appendix C.

Table 6

EXAMPLE OF AN EIGENVECTOR INPUT FILE FOR NEIGN=9

0	
231.34	68.89
37.39	72.13
220.68	-147.67
174.68	-24.78
1027.02	-530.07
10	
-20.61	-5.43
-7.41	-4.91
-14.22	6.77
-5.21	0.47
-93.51	21.15
9	
34.02	-15.01
-0.10	-22.52
19.54	-8.31
16.46	-17.55
-27.96	-79.37
8	
22.65	-22.08
46.91	-50.71
38.16	-27.88
-3.64	-7.96
-2.89	42.80
7	
80.61	-15.48
-27.14	32.29
-7.77	0.91
-28.79	3.24
-8.43	22.63

Table 7

WIND FIELD SOLUTION INPUT DATA FILES

Record	Parameters	Format	Remarks
1	JSITE, IV, DNI, AVTHK, SLFAC	1X, I4, 2I5, F10.0, F5.1	JSITE, DNI and SLFAC are not used. AVTHK is the mixing depth (m) used to obtain solutions
2-51	$(U,V,W,Z)_{sfc}$	8F10.2	The wind components and elevations at the bottom (rec 2-51) and top (52-101) of the layer for the mean solution. Values from this file are stored in the arrays PRECAL (IX, IY, IZ, ICOMP,LEVEL, NEIGN) and RHS (IX, IY, LEVEL) by Subroutine WINDDO. Sets of values are read for rows of grid points (west-to-east). The rows are read south-to-north.
52-101	$(U,V,W,Z)_{top}$	AF10.2	
102	same as record 1	same as record 1	
103-202	same as 2-101	same as 2-101	same as 2-101 except for the next eigen-vector solution
repeats through all solutions			solutions for eigenvectors of decreasing importance.

3. Interactive Inputs

When the program is run, it will ask for a series of inputs. Examples are given in Appendix C. The remainder of this section discusses the required interactive inputs.

(1) Receptor spacing (m)? The user supplies the distance in meters between points at which concentrations are to be calculated. The area within which concentration is calculated will be a square twenty times this size centered on the source. The spacing number should probably be between 25 m and 500 m in order to provide reasonable coverage and detail. Puffs are tracked outside this area--to about ±15 km in all directions, and can return to the calculation grid upon wind reversal.

(2) How many wind sites? This input must match the number of wind measurement sites used for the calculation of the wind field solution sets (excluding the upper level wind).

(3) How many empirical orthogonal functions? The number input here must be $\leq$ than the number of eigenvector solution fields that you have provided.

(4) Source x, y, z? The input values for x and y should be in the same coordinate system as the calculation grid (i.e. the origin is at -15000, -15000) and should be near the center of the computational area (near 0,0). The height (z) should be relative to the local surface. The source should be below the mixing height. The input values for x, y and z are separated by commas.

(5) Hour? A 2-digit (24-hr clock) hour of the day is input. This is a bookkeeping measure and no checks are made for consistency (e.g. with stability class).

(6) Mixing height and stability class (Pasquill-Gifford, as an integer from 1 to 6)? These values remain unchanged through the next hour. Separate the input values with commas.

(7) Source strength (grams/sec)? Almost any reasonable value can be used. The value remains constant for the ensuing hour. Calculated concentration values will be in terms of whatever mass units are used (per cubic meter).

(8) Speed (m/s) and direction? (Degrees clockwise from north--meteorological convention). This input request will be repeated enough times to provide inputs at all wind sites and at the top of the domain for the beginning of the hour. After asking for the winds at the beginning of the hour, the program asks for the values at the end of the hour. This second set of inputs may differ in any reasonable way from those that have been input for the beginning of the hour. Separate the speed and direction inputs with a space or comma. Zero wind speeds are permissible.

The program calculations take place after the winds have been input. At the end of the hour it calculates concentrations, returns to step (6), and asks for the mixing height and stability for the coming hour. Program execution can be terminated with the input of a negative mixing depth. It will then write the calculated concentration values and stop. If you continue , the program then asks for source strength for the next hour, followed by a request for winds at all sites for the end of the next hour. The program will repeat the process until five hours have elapsed (or a negative mixing height is supplied). It then stops and writes the accumulated results.

REFERENCES

Bhumralkar, C.M., R.L. Mancuso, F.L. Ludwig, and D.S. Renne', 1980: "A Practical and Economic Method for Estimating Wind Characteristics at Potential Wind Energy Conversion Sites, "Solar Energy, 25, 55-65.

Chen, T.C. 1980: "On the Energy Exchange Between the Divergent and Rotational Components of Atmospheric Flow over the Tropics and Subtropics at 200 mb During Two Northern Summers," Mon. Wea. Rev., 108, 896-912.

Endlich, R.M., 1967: "An Interactive Method for Altering the Kinematic Properties of Wind Fields," J. Appl. Meteorol., 6, 837-844.

Endlich, R.M., 1984: "Wind Energy Estimates by Use of a Diagnostic Model," Boundary Layer Meteorol., 30, 375-386.

Endlich, R.M. and J.D. Lee, 1983: "An Improved Diagnostic Model for Estimating Wind Energy," Final Report to Pacific Northwest Laboratory, Subcontract B-D5789-A-E, SRI International, Menlo Park, California 94025.

Endlich, R.M., F.L. Ludwig, C.M. Bhumralkar, and M.A. Estoque, 1982: "A Diagnostic Model for Estimating Winds at Potential Sites for Wind Turbines," J. Appl. Meteorol., 21, 1441-1454.

Goodin, W.R., G.J. McRae, and J.H. Seinfield, 1980: "An Objective Analysis Technique for Constructing Three-Dimensional Urban-Scale Wind Fields," J. Appl. Meteorol., 19, 98-108.

Liu, C.Y. and W.R. Goodin; 1976: "An Iterative Algorithm for Objective Wind Field Analysis," Mon. Wea. Rev., 104, 784-792.

Ludwig, F.L. 1982: "Effect of a Change in Atmospheric Stability on the Growth Rate of Puffs Used in Plume Simulation Models," J. Appl. Meteorol., 21, 1371-1373.

Ludwig, F.L., 1984: "Transport and Diffusion Calculation During Time and Space-varying Meteorological Conditions Using a Microcomputer," J. Air Poll. Cont. Assoc., 31, 227-232.

Ludwig, F. L., and G. Byrd, 1980. "An Efficient Method for Deriving Mass-Consistent Flow Fields from Wind Observations in Rough Terrain," Atmos. Environ., 14, 585-587.

Ludwig, F.L., L.S. Gasiorek, R.E. Ruff, 1977: "Simplification of a Gaussian Puff Model for Real-Time Minicomputer Use," Atmos. Environ., 11, 431-436.

Sasaki, Y., 1958: "An Objective Analysis Based on the Variational Method," J. Meteorol. Soc. Japan, 36, 77-88.

Sasaki, Y., 1970a: "Some Basic Formalisms in Numerical Variational Analysis," Mon. Wea. Rev., 98, 875-883.

Sasaki, Y., 1970b: "Numerical Variational Analysis Formulated Under the Constraints Determined by Longwave Equations and Lowpass Filter," Mon. Wea. Rev., 98, 884-898.

Sherman, C.A. 1978: "A Mass-Consistent Model for Wind Fields over Complex Terrain," J. Appl. Meteorol., 17, 312-319.

Slade, D. (ed.), 1968: Meteorology and Atomic Energy, U. S. Atomic Energy Comm. (TID 24190), Oak Ridge, TN, 445 pp.

Turner, D.B., 1970: "Workbook of Atmospheric Dispersion Estimates," Public Health Service Publication 999-AP-26.

Appendix A

PROGRAM LISTINGS FOR WIND PREPROCESSORS

```
      PROGRAM TERRAIN(INPUT,OUTPUT,TAPE1,TAPE2,TAPE5=INPUT)
C**FOR READING DIGITAL TERRAIN DATA AT .005 DEG INTERVALS AND SMOOTHING
C  TO GET .01 DEG VALUES
C**TAPE 1 IS INPUT FROM NCIC, TAPE 2 IS SMOOTHED OUTPUT
C* PT(1,1) IS AT SW CORNER OF 1 DEG BLOCK. DATA EXTEND TO NEXT WHOLE
C  LAT AND LONG.
C* TERRAIN HEIGHT DATA ARE HT(KY,JX). KY IS COUNTER S TO N, JX IS W TO E
C** BY R. ENDLICH , SRI, MAY 79
      DIMENSION  IX(2),IY(2)
      DIMENSION  IA(1800),IUNPK(14)
      DIMENSION KHAR(10),LINE(12)
      COMMON /C1/ RT(4370)
      COMMON /C2/ PR(202,2)
      COMMON /C3/ HT(101,101)
      COMMON /C4/ IXSWC, DELX
      DATA KHAR/10H 123456789,10H ABCDEFGHI,10H JKLMNOPQR,10H STUVWXYZ+,
     +10H 123456789,10H ABCDEFGHI,10H JKLMNOPQR,10H STUVWXYZ+,10H 123456
     +789,10H ABCDEFGHI/
C       READ 9010, NMAREA,TAPEID,SECTN,IFILE
C       PRINT 9015,NMAREA,TAPEID,SECTN,IFILE
        REWIND 1
      NUMREC = 0   $  JX = 1    $  K12 = 2
C READ HEADER RECORD
 200    BUFFER IN (1,1) (RT(1),RT(4370))
        ERROR=1H     $  NUMREC = NUMREC +1
        IF (UNIT(1))  230,210,220
 210     PRINT 22
        CALL REMARK(20HEOF AT HEADER RECORD)
       GO TO 600
 220    ERROR =4HP.E.
C  HEADER RECORD IS OK
 230    LAST = 4370
        LENREC = LENGTH(1)
        K12 = 3 - K12
        IF (NUMREC .EQ. 1) CALL APRINT(RT,NAME,IFILE,8)
C       PRINT 331, NUMREC, LENREC, ERROR, (RT(I),I=1,LENREC)
       IBYTE = 9  $  IF (NUMREC .EQ.1) IBYTE = 91
       CALL UNPK8(RT,IBYTE ,IUNPK(1),14)
       IX(K12)=SHIFT(IUNPK(1),8) .OR.IUNPK(2)
       IY(K12) = SHIFT(IUNPK(3),8) .OR.IUNPK(4)
       IDX = SHIFT(IUNPK(5),8) .OR.IUNPK(6)
       IDY = SHIFT(IUNPK(7),8) .OR.IUNPK(8)
       NPTS     = SHIFT(IUNPK(9),8) .OR.IUNPK(10)
       IH1  = SHIFT(IUNPK(11),8) .OR.IUNPK(12)
       IH2  =  SHIFT(IUNPK(13),8) .OR.IUNPK(14)
      PRINT 332, NUMREC,LENREC,IX(K12),IY(K12),IDX,IDY,NPTS     ,IH1,IH2
     +      ,ERROR
       IF (JX .LT. 2 .AND. NPTS .LT. 1740) GO TO 200
        IF (NPTS .LT. 1740) GO TO 235
       CALL UNPK16(RT,(IBYTE+10)/2+1,IA(1),NPTS)
C*     PRINT 9027, (IA(L),L=1701,1800)
C*     PRINT 9027, (IA(L),L=   1, 100)
       CALL SELECT(NPTS,K12,(IBYTE+10)/2)
        IF (JX .GT. 1)  GO TO 245
  235   CONTINUE
```

```
            DO 240 KY1 = 1,202
              PR(KY1,3-K12) = PR(KY1,K12)
  240       CONTINUE
              GO TO 250
  245     CONTINUE
        NSX = IXSWC + (JX -1) * DELX
        PRINT 9026, NUMREC,JX,K12,IX(K12),NSX
         IF (NSX .GT. IX(K12) ) GO TO 200
C*      PRINT 9026, NUMREC,JX,K12,IX(K12),NSX
  250   CALL AVERG(JX)
C       PRINT 9028, (HT(KY,JX),KY=91,101)
         JX = JX + 1
          IF (JX .GT. 101) GO TO 600
        GO TO 200
 600     CONTINUE
 602     IF (JX .GT. 101) GO TO 605
           DO 603 KY = 1,101
             HT(KY,JX) = HT(KY,JX-1)
 603       CONTINUE
           JX = JX +1
         GO TO 602
 605     CONTINUE
       WRITE(2) ((HT(K,J),K=1,101),J=1,101)
       READ(5,15) HTINT
        PRINT 9032, HTINT
          SCFC = 1.0/HTINT
        PRINT 9040
        DO 615 KY = 1,101
         KYP = 101 - KY +1
        DO 610 JX = 1,101
         HTD = HT(KYP,JX)
         K = SCFC * HTD +1
         IF (HTD .GT. 0.0 .AND. HTD .LE. 25.0) K =32  ; FOR SHORELINE
         IF (K .LT.1) K =1 $  IF (K .GT. 100) K = 100
         CALL CHAR(KHAR,K,LINE,JX,1)           ; USUAL ORDER REVERSED
  610   CONTINUE
         PRINT 9030,KYP,LINE
  615   CONTINUE
        PRINT 9040
 10    FORMAT (A10)
 15    FORMAT (F6.1)
 21    FORMAT (*OERROR 110 - CANT RECOGNIZE NAME*A11,/,(10X,10A11))
 22    FORMAT (*OERROR 210 - EOF AT HEADER RECORD*)
 23    FORMAT (2X,I6,2A10,I6)
 331   FORMAT (*ORECORD *2I5,A12,/,(10X,5022))
 332   FORMAT (1X,I4,I6,2X,2I6,2X,2I6,I7,2X,2I6,5X,A10)
 9010 FORMAT  (3A10,I5)
 9015 FORMAT (2X,3A10,* FILE =*I6)
 9026 FORMAT (2X,6I5)
 9027 FORMAT  (1X,4(2X,5I5))
 9028 FORMAT (1X,4(1X,5F6.0))
 9030 FORMAT (4X,I5,2X,12A10)
 9032 FORMAT (1H1,*  HEIGHT INTERVAL = *F6.1)
 9040 FORMAT(/11X,9(1H0),10(1H1),10(1H2),10(1H3),10(1H4),10(1H5),
     +10(1H6),10(1H7),10(1H8),10(1H9),1H0/11X,
      +10(10H1234567890)/)
  9042 FORMAT (100(1X,30(1X,O3)/))
           END
```

```
      SUBROUTINE AVERG(JCOL)
C* AVERAGE PR(X,1) AND PR(X,2) TO GIVE SMOOTHED TERRAIN AT .01 DEG INCR
      COMMON /C2/ PR(202,2)
      COMMON /C3/ HT(101,101)
         HT(1,JCOL) = 0.5 *(PR(1,1) + PR(1,2))
       DO 100  KY2 = 3,201,2
         KY1 = (KY2 +1)/2
         HT(KY1,JCOL) = .16667 * (PR(KY2-1,1) +PR(KY2,1) + PR(KY2+1,1)
     +                     + PR(KY2-1,2) + PR(KY2,2) + PR(KY2+1,2))
  100  CONTINUE
       RETURN
       END


      SUBROUTINE SELECT(NN,JCOL,I16)
C* IUNPK IS RAW TERRAIN DATA, NN IS NO. OF PTS IN PROFILE, JCOL IS COL.
      COMMON /C1/ RT(4370)
      COMMON /C2/ PR(202,2)
         CALL UNPK16(RT,I16+1,IB,1)
         PR(1,JCOL) = IB
         DELY = (NN-1)/200.0
         KBEG = 2    $  KEND = 202   $  KSKIP = 1
       DO 50 KN = KBEG,KEND,KSKIP
         NSY = 1.0 + (KN-1) * DELY
         IF (NSY .GT. NN) GO TO 45
         CALL UNPK16(RT,I16+NSY,IB,1)
         PR(KN,JCOL) = IB
         GO TO 48
 45      PR(KN,JCOL) = PR(KN-1,JCOL)
C48     IF (KN .LT.  6 .OR. KN .GT. 195) PRINT 9002, KN,NSY,PR(KN,JCOL)
 48     CONTINUE
 50     CONTINUE
 9002 FORMAT (3X,2I8,F10.0)
       RETURN
       END


      SUBROUTINE UNPK16(PACK,IB,I16,N16)
C         REVISION DATE:  JULY 24, 1978
C  UNPACK *N16* 16-BIT BYTES FROM *PACK*, STARTING WITH THE IB-TH ONE,
C    AND STORE INTO *I16*.
C  USES *UNPK8* TO FIRST UNPACK INTO 8 BIT BYTES.
C-----------------------------------------------------------------------
      DIMENSION PACK(1),I16(1),I8BB(2)
C-----------------------------------------------------------------------
         I8  =  2*IB - 3
         DO 10  I=1,N16
           I8  =  I8 + 2
           CALL UNPK8(PACK,I8,I8BB,2)
        I16(I) = SHIFT(I8BB(1) .AND. 177B,8) .OR. I8BB(2)
C          CHECK FOR SIGN BIT
           IF ((I8BB(1).AND.200B) .EQ. 0)  GO TO 10
        I16(I) = -I16(I) -1
   10    CONTINUE
         RETURN
C-----------------------------------------------------------------------
      END
```

```
      SUBROUTINE APRINT(DATA,NAME,IFIL,IBEG)
C          REVISION DATE:  APRIL 24, 1979
C   PRINT THE CONTENTS OF A TYPE A LOGICAL RECORD FROM A DIGITAL TERRAIN TAPE
C     USES UNPK8 AND CHAR (FROM LIBRARY WINDLIB/UN=DEBJF)
C-----------------------------------------------------------------------
      LOGICAL DEBUG
      DIMENSION DATA(1),IUNPK(16),CORNER(4),SHEET(2),UNITS(2)

      COMMON /C4/ IXSWC, DELX
      DATA  DEBUG/.FALSE./      ;NO DEBUGGING PRINTOUT
      DATA  DEBUG/.TRUE./       ;DELETE THIS CARD WHEN THIS SUBROUTINE WORKS
      DATA  UNITS/6HMETERS,5H FEET/
      DATA  CORNER/9HSOUTHWEST,9HNORTHWEST,9HNORTHEAST,9HSOUTHEAST/
C-----------------------------------------------------------------------
      N32BIT(I1,I2,I3,I4)  =  SHIFT(SHIFT(I1,52).OR.SHIFT(I2,44)
     +                         .OR.SHIFT(I3,36).OR.SHIFT(I4,28),-28)
C-----------------------------------------------------------------------
    1      FORMAT (" SUBAREA"A11"FILE:"I2)
    2      FORMAT (35X," MAP   PROJ   ELEVATION"
     +        /3X"S H E E T    SERIES  EDITION   PROJ   ZONE   U N I T S"
     +        /"   N U M B E R"5X"I D       I D     CODE    NUM   CODE TYPE"
     +        /1X,A10,A2,3X,A6,3X,A6,3X,I4,3X,I4,2X,I4,1X,A6)
    3      FORMAT (13X"I N C H E S   D E G R E E S"7X"DEG.MIN.SEC"
     +            ,11X"ARC-SECONDS"
     +            /15X"X      Y"8X"LON   LAT"6X"L O N     L A T"
     +            ,8X"L O N  L A T")
    4      FORMAT (1X,A9,2X,2F6.2,3X,F7.2,F6.2,3X,I4"."I2"."I2
     +            ,I3"."I2"."I2,4X,2I7)
C-----------------------------------------------------------------------
C     BYTES 1-12:  SHEET NUMBER (EBCDIC CODE)
   10      CALL UNPK8(DATA,IBEG+1,IUNPK,12)
   11      FORMAT ("OBYTES  1-12:"12(1X,O3))
           IF (DEBUG)  PRINT 11, (IUNPK(I),I=1,12)
           DO 12  I=1,12
             CALL KEB264(IUNPK(I),SHEET,I)
   12      CONTINUE
C     BYTES 13-18:  SERIES ID (EBCDID CODE)
   20      CALL UNPK8(DATA,IBEG+13,IUNPK,6)
   21      FORMAT ("OBYTES 13-18:"12(1X,O3))
           IF (DEBUG)  PRINT 21, (IUNPK(I),I=1,6)
           DO 22  I=1,6
             CALL KEB264(IUNPK(I),SERIES,I)
   22      CONTINUE
C     BYTES 19-24:  EDITION ID (EBCDIC CODE)
   30      CALL UNPK8(DATA,IBEG+19,IUNPK,6)
   31      FORMAT ("OBYTES 19-24:"12(1X,O3))
           IF (DEBUG)  PRINT 31, (IUNPK(I),I=1,6)
           DO 32  I=1,6
             CALL KEB264(IUNPK(I),EDITION,I)
   32      CONTINUE
C     MAP PROJECTION CODE, PROJECTION ZONE NUMBER, AND ELEVATION UNITS CODE
   40      CALL UNPK8(DATA,IBEG+25,IUNPK,6)
   41      FORMAT ("OBYTES 25-30:"12(1X,O3))
           IF (DEBUG)  PRINT 41, (IUNPK(I),I=1,6)
           IPROJ   =  SHIFT(IUNPK(1),8).OR.IUNPK(2)     ;BYTES 25-26
```

```
          IZONE   =  SHIFT(IUNPK(3),8).OR.IUNPK(4)      ;BYTES 27-28
          IUNITS  =  SHIFT(IUNPK(5),8).OR.IUNPK(6)      ;BYTES 29-30
C    BYTES 31-46:   (X,Y) OF MAP CORNERS -- INCHES
   50     CALL UNPK8(DATA,IBEG+31,IUNPK,16)
   51     FORMAT ("OBYTES 31-46:"16(1X,O3))
          IF (DEBUG)  PRINT 51, (IUNPK(I),I=1,16)
          XSW = 0.01*FLOAT((SHIFT(IUNPK( 1),8).OR.IUNPK( 2)))
          YSW = 0.01*FLOAT((SHIFT(IUNPK( 3),8).OR.IUNPK( 4)))
          XNW = 0.01*FLOAT((SHIFT(IUNPK( 5),8).OR.IUNPK( 6)))
          YNW = 0.01*FLOAT((SHIFT(IUNPK( 7),8).OR.IUNPK( 8)))
          XNE = 0.01*FLOAT((SHIFT(IUNPK( 9),8).OR.IUNPK(10)))
          YNE = 0.01*FLOAT((SHIFT(IUNPK(11),8).OR.IUNPK(12)))
          XSE = 0.01*FLOAT((SHIFT(IUNPK(13),8).OR.IUNPK(14)))
          YSE = 0.01*FLOAT((SHIFT(IUNPK(15),8).OR.IUNPK(16)))
      DELX = XSE - XSW
          IXSWC = 100 * XSW
C    BYTES 47-78:   LON-LAT OF CORNERS OF AREA COVERED BY THE MAP
   60     CALL UNPK8(DATA,IBEG+47,IUNPK,16)
   61     FORMAT ("OBYTES 47-62:"16(1X,O3))
   62     FORMAT ("OBYTES 63-78:"16(1X,O3))
          IF (DEBUG)  PRINT 61, (IUNPK(I),I=1,16)
          LONSW  =  N32BIT(IUNPK( 1),IUNPK( 2),IUNPK( 3),IUNPK( 4)) -1
          LATSW  =  N32BIT(IUNPK( 5),IUNPK( 6),IUNPK( 7),IUNPK( 8))
          LONNW  =  N32BIT(IUNPK( 9),IUNPK(10),IUNPK(11),IUNPK(12)) -1
          LATNW  =  N32BIT(IUNPK(13),IUNPK(14),IUNPK(15),IUNPK(16))
          CALL UNPK8(DATA,IBEG+63,IUNPK,16)
          IF (DEBUG)  PRINT 62, (IUNPK(I),I=1,16)
          LONNE  =  N32BIT(IUNPK( 1),IUNPK( 2),IUNPK( 3),IUNPK( 4)) -1
          LATNE  =  N32BIT(IUNPK( 5),IUNPK( 6),IUNPK( 7),IUNPK( 8))
          LONSE  =  N32BIT(IUNPK( 9),IUNPK(10),IUNPK(11),IUNPK(12)) -1
          LATSE  =  N32BIT(IUNPK(13),IUNPK(14),IUNPK(15),IUNPK(16))
          PRINT 1, NAME,IFIL
          PRINT2,SHEET,SERIES,EDITION,IPROJ,IZONE,IUNITS,UNITS(IUNITS+1)
          PRINT 3
          X=XSW $ Y=YSW $ LON=LONSW $ LAT=LATSW $ IGOTO=1   $  GO TO 80
   71     X=XNW $ Y=YNW $ LON=LONNW $ LAT=LATNW $ IGOTO=2   $  GO TO 80
   72     X=XNE $ Y=YNE $ LON=LONNE $ LAT=LATNE $ IGOTO=3   $  GO TO 80
   73     X=XSE $ Y=YSE $ LON=LONSE $ LAT=LATSE $ IGOTO=4   $  GO TO 80
   74     RETURN
C    PRINT A LINE FOR THE LOCATIONS OF A CORNER
   80     DLON   =  LON/3600.0    $   DLAT = LAT/3600.0
          LOND   =  ISIGN(IABS(LON)/3600,LON)
          LATD   =  LAT/3600
          LONM   =  MOD((IABS(LON)/60),60)
          LATM   =  MOD(LAT/60, 60)
          LONS   =  MOD(IABS(LON), 60)
          LATS   =  MOD(LAT,60)
     CALL DATEX(DATE1,DATE2)
     PRINT 9005, DATE1,DATE2
 9005 FORMAT(1X,A10,A2)
          PRINT 4, CORNER(IGOTO),X,Y,DLON,DLAT,
     +             LOND,LONM,LONS, LATD,LATM,LATS, LON,LAT
     WRITE(2) DATE1,DATE2,CORNER(IGOTO),X,Y,DLON,DLAT,
     1          LOND,LONM,LONS,LATD,LATM,LATS,LON,LAT
          GO TO (71,72,73,74),IGOTO
```

```
      SUBROUTINE GO2EOF(LUN,NEOF,NREC)
C           REVISION DATE:  AUGUST 1, 1977
C     SKIP TO THE NEOF-TH END OF FILE ON LOGICAL UNIT NUMBER *LUN*.
C       NOTE THAT THIS VERSION USES *BUFFER IN* TO READ THE DATA.

   10       IF (NEOF .LT. 1)  GO TO 90
            IEOF  =  0

   20       BUFFER IN (LUN,1)  (NULL,NULL)
            NREC  =  NREC + 1
            IF (UNIT(LUN))  20,30,20

   30       IEOF  =  IEOF + 1
            IF (IEOF .LT. NEOF)  GO TO 20

   90       RETURN

      END
```

```
      SUBROUTINE KEB264(I8BIT,STRING,K)
C           REVISION DATE:  JULY 21, 1978
C   BY TABLE LOOKUP, CONVERT *I8BIT* FROM AN 8-BIT, RIGHT JUSTIFIED, EXTERNAL BCD
C     CODED (EBCDIC) CHARACTER TO A CDC-6400 INTERNAL DISPLAY CODED CHATACTER
C     STORED IN THE PROPER CHARACTER POSITION OF *STRING*.
C   USE SUBROUTINE *CHAR* TO TRANSFER THE PROPER SIX BIT CHATACTER FROM THE
C     PACKED CONVERSION TABLE INTO THE K-TH CHARACTER OF *STRING*.
C-------------------------------------------------------------------------
      DIMENSION TABLE(26)
      DATA  TABLE/
     +      0          ,  0          ,  0          ,  0
     +   ,  0          ,  0          ,  0          ,10H     .<(+
     +   ,  0          ,10H $*);¬-/  ,10H       ,  ,10H>
     +   ,10H  :  "=" A,10HBCDEFGHI  ,10H     JKLMN,10HOPQR
     +   ,10H  STUVWXYZ,  0          ,  0          ,10H   ABCDEFG
     +   ,10HHI       J,10HKLMNOPQR  ,10H      STUV,10HWXYZ
     +   ,10H0123456789,  0                                   /
C-------------------------------------------------------------------------
         CALL CHAR(TABLE,I8BIT+1,STRING,K,1)
         RETURN
C-------------------------------------------------------------------------
      END
```

```
      PROGRAM  GRIDHT(INPUT,OUTPUT,TAPE1,PUNCH)
C* USE SMOOTHED .01 DEG TERRAIN HEIGHTS. PICKS OUT PROPER VALUES
C  FOR GRID PTS.  SMOOTHES FURTHER FOR AREAS EQUIVALENT TO GRID SPACING
C* BY R  ENDLICH, SRI 7/79
     DIMENSION ALAT(22,22),ALNG(22,22),GHT(22,22),GX(22,22),GY(22,22)
     DIMENSION HT(101,101)
     READ 9001, MX,NY,NGCX,NGCY,GINCX,GINCY
     PRINT 9002, MX,NY,NGCX,NGCY,GINCX,GINCY
      READ   9003, SLAT, SLNG, SHGT
      PRINT  9004, SLAT, SLNG, SHGT
C*  HEIGHT DATA ARE HT(IX,IY) IN FEET
C* NGCX, NGCY ARE COLUMN AND ROW OF SITE.  IX GOES 1,MX  IY GOES 1,NY
C   (1,1) IS SW CORNER.  GRID INCREMENTS AREIN KM.
C*  GX( , ) HAS X VALUES OF GRID PTS.  X=0 IS AT NGCX, THE SITE.
C* COMPUTE LAT AND LONG OF GRID PTS
      DO 100  IX =1,MX   $  DO 100 IY = 1,NY
        GX(IX,IY) = (IX -NGCX) * GINCX
        GY(IX,IY) = (IY - NGCY) * GINCY
        ALAT(IX,IY) = SLAT + GY(IX,IY)/111.0
        COSS = COS(SLAT/57.2958)
        ALNG(IX,IY) = SLNG + GX(IX,IY)/(111.0 * COSS)
        IF (IX .EQ. 1 .OR. IX .EQ. MX)
     +  PRINT 9010,GX(IX,IY),ALNG(IX,IY),GY(IX,IY),ALAT(IX,IY)
 100  CONTINUE
          CALL MEMSETX(0.0,GHT,22*22)
     READ 9001, NFILES
      PRINT 9006, NFILES
      NCOUNT = 0
 140  CONTINUE
C*  READ AREA LIMITS AND THE 0.01 DEG HEIGHTS
      PRINT 3
      DO 160 IC = 1,4
 150  CONTINUE
      READ (1) DATE,DATE2,CORNER,X,Y,DLON,DLAT,LOND,LONM,LONS,LATD,
     +  LATM,LATS,LON,LAT
       IF (EOF(1))  150,155
 155   CONTINUE
      PRINT 4, CORNER,X,Y,DLON,DLAT,LOND,LONM,LONS,LATD,LATM,LATS,LON,
     +    LAT
       IF (IC .GT. 1) GO TO 160
       WLNG= DLON
       ASLAT = DLAT
       ELNG = WLNG + 1.0
       ANLAT = ASLAT + 1.0
 160  CONTINUE
      PRINT 9014, ANLAT,ASLAT,ELNG,WLNG
      NCOUNT = NCOUNT + 1
      READ (1) ((HT(K,J),K=1,101),J=1,101)
C*  HT(1,1) IS AT SW CORNER.  K INCREASES TO N. J INCREASES TO EAST.
       DO 165  J =1,5
 165  PRINT 9012,  (HT(K,J),K=1,20)
C* SEARCH THE AREA TC SFE IS A GRID PT IS ENCLOSED.  IF IT IS PICK
C   THE HEIGHT VALUE.
      DO 200 IX =1,MX   $  DO 200 IY = 1,NY
        IF (ALAT(IX,IY) .GT. ANLAT .OR. ALAT(IX,IY) .LT. ASLAT) GO TO
```

```
     + 200
          IF (ALNG(IX,IY) .GT. ELNG .OR. ALNG(IX,IY) .LT. WLNG) GO TO
     +  200
          AX =  (ALNG(IX,IY) -WLNG) + .005
          LX = 100 * AX
          AY = (ALAT(IX,IY) - ASLAT) + .005
          LY = 100 * AY
C*  LX AND LY ARE INCICES TO PICK OUT HT( ) VALUES
C* COMPUTE SMOOTHING INTERVAL
          IS = GINCY/(2.5 * 1.11)
          LYN = LY + IS
          IF (LYN .GT. 101) LYN = 101
          LYS = LY - IS
          IF (LYS .LT. 1) LYS = 1
          LXE = LX + IS
          IF (LXE .GT. 101) LXE = 101
          LXW = LX - IS
          IF (LXW .LT. 1) LXW = 1
          GHT(IX,IY) = 0.2 * (HT(LY,LX) + HT(LYN,LX) + HT(LYS,LX) +
     +      HT(LY,LXE) + HT(LY,LXW))
          IF (IX .EQ.1 .OR. IX .EQ. MX)
     + PRINT 9021, AX,LX,AY,LY,GHT(IX,IY),IS,IX,IY
  200  CONTINUE
       PRINT 9015
       DO 220 IY = 1,NY
        IP = NY + 1 - IY
  220  PRINT 9012, (GHT(IX,IP),IX=1,MX)
          IF (NCOUNT .LT. NFILES) GO TO 140
C* INSERT HEIGHT AT SITE
       GHT(NGCX,NGCY) = SHGT
C* PUNCH THE SOUTHERN ROW FIRST FOR USE BY TOPO
       DO 225 IY = 1,NY
  225  PUNCH 9016, (GHT(IX,IY),IX=1,MX)
    3      FORMAT (13X"I N C H E S   D E G R E E S"7X"DEG.MIN.SEC"
     +             ,11X"ARC-SECONDS"
     +             /15X"X      Y"8X"LON   LAT"6X"L O N     L A T"
     +             ,8X"L O N  L A T")
     *     FORMAT (1X,A9,2X,2F6.2,3X,F7.2,F6.2,3X,I4"."I2"."I2
     +             ,I3"."I2"."I2,4X,2I7)
 9001 FORMAT  (4I5,2F6.1)
 9002 FORMAT (/*  NO. PTS W-E =*I3,*  N-S =*I3,* SITE COL =*I3,* SITE
     +ROW =*I3,* GRID INCREMENTS X =*F5.1,* Y =*F5.1)
 9003 FORMAT  (4F10.3)
 9004 FORMAT (/*  SITE LAT =*F8.2,*  LONG =*F8.2,* HEIGHT IN FT=*F9.0)
 9006 FORMAT (/*   NO. OF FILES FOR 1 BY 1 DEG AREAS =*I5)
 9010 FORMAT (10F10.2)
 9012 FORMAT (20F6.0)
 9013 FORMAT (/20F6.1)
 9014 FORMAT (/*   NORTHERN LAT =*F6.1,*  SOUTHERN LAT = *F6.1,* EASTERN
     + LNG =*F7.1,* WESTERN LONG =*F7.1/)
 9015 FORMAT  (/*    HEIGHT VALUES AT GRID POINTS*/)
 9016 FORMAT (11F6.0)
 9021 FORMAT (/2(F8.3,I4),F10.1,3I4)
      STOP
      END
```

```
      PROGRAM GEOCAL (INPUT,OUTPUT,TAPE1,TAPE2,TAPE5=INPUT,TAPE6=
     *OUTPUT)
C
C           THIS PROGRAM CREATES A DATA FILE OF WEATHER DATA FOR GROUPS
C      OF WEATHER STATIONS IN VARIOUS U.S. AREAS DURING 1977-1978.  THE
C      GEOSTROPHIC WINDS AND STABILITY INDICES ARE ALSO COMPUTED.
C
      COMMON /REC/ ID(80),NF,NR,NP,LEN,IEOF
      COMMON /CSTAB/SP180,CP180,DAY
      COMMON /DAT/ SP(10),ALAT(10),ALON(10),IU1,IU2,IU3,C1,C2,FC,
     1AVLAT,AVLON,DENOM,STLT1,STLT2,STLT3,STLN1,STLN2,STLN3,COSLAT
      DIMENSION ISTA(10),STAN(10),IGMT(10),SI(10),WD(10),WS(10),OC(10),
     1SPS(10),WDS(10),OCS(10),WSS(10),U(10),V(10),P180S(10),P180C(10),
     2IFMT2(6),IFMT5(4),STANS(10),US(10),VS(10),SIS(10),MODA(12)
      DATA (MODA=31,28,31,30,31,30,31,31,30,31,30,31)
      DATA ACR /0.0174533/
      DATA SPS,WDS,WSS,OCS/10*1013.0,10*270.0,10*1.0,10*5.0/
      DATA US,VS,SIS,UGS,VGS /10*1.0,10*1.0,10*3.0,0.0,0.0/
 1    FORMAT (1H1,10X*CREATION OF A DATA FILE OF SURFACE METEOROLOGICAL
     1DATA, GEOSTROPHIC WINDS AND STABILITY INDICES.*/)
 3    FORMAT (10I8)
 4    FORMAT (10F8.0)
 6    FORMAT (A2,I1,R7)
 7    FORMAT (A9,I1)
 8    FORMAT (/1H ,*END OF FILE, TAPE2 ... NO. OF RECORDS =*I5)
 9    FORMAT (3X*SFC P:*5(F6.1,17X))
 10   FORMAT (8A10)
 11   FORMAT (1H ,8A10)
 12   FORMAT (A2)
 13   FORMAT (8X,A2)
 14   FORMAT (1X,A4)
 15   FORMAT (8X,A1)
 16   FORMAT (1H ,*WS MISSING FOR SITE*I6,* ON*I7,I2)
 17   FORMAT (1H ,*WD MISSING FOR SITE*I6,* ON*I7,I2)
 18   FORMAT (1H ,*SP MISSING FOR SITE*I6,* ON*I7,I2)
 19   FORMAT (1H ,*OC MISSING FOR SITE*I6,* ON*I7,I2)
C
C     INPUT ...
C
C     IFMT2 = HEADER FORMAT FOR DATA PRINTOUT.
C     IFMT5 = DATA FORMAT FOR PRINTOUT.
C     NSTA = NO. OF STATIONS IN THE AREA.
C     IU1, IU2, IU3 = INDICIES OF STATIONS FOR GEOS WIND COMPUTATIONS.
C     ISTA = STATION NUMBERS.
C     IGMT = GMT TIME CORRECTION  FOR EACH STATION.
C     ALAT = LATITUDE FOR EACH STATION
C     ALON = LONGITUDE  FOR EACH STATION.
C     IDATES = STARTING DATE OF INTEREST
C     IGHS = STARTING GMT HOUR.
C
      READ (5,3) NSTA,IU1,IU2,IU3,IDATES,IGHS
      WRITE (6,3) NSTA,IU1,IU2,IU3,IDATES,IGHS
      READ (5,3) (ISTA(L),L=1,NSTA) $ READ (5,3) (IGMT(L),L=1,NSTA)
      WRITE(6,3) (ISTA(L),L=1,NSTA) $ WRITE(6,3) (IGMT(L),L=1,NSTA)
      READ (5,4) (ALAT(L),L=1,NSTA) $ READ (5,4) (ALON(L),L=1,NSTA)
      WRITE (6,4) (ALAT(L),L=1,NSTA) $ WRITE (6,4) (ALON(L),L=1,NSTA)
      READ (5,10) IFMT2 $ WRITE (6,11) IFMT2
      READ (5,10) IFMT5 $ WRITE (6,11) IFMT5
      ENCODE (10,6,IFMT2(2))IFMT2(2),NSTA,IFMT2(2)
      ENCODE (10,7,IFMT5(1)) IFMT5(1),NSTA
      WRITE (6,1) $ WRITE (6,IFMT2) $ IS=NRW=0
```

```
      DO 50 I=1,NSTA
      P180=ACR*ALAT(I)
      P180S(I)=SIN(P180) $ P180C(I)=COS(P180)
      STANS(I)=ISTA(I)
 50   CONTINUE
      C1=1.0 $ RHO=1.1 $ C2=100.0/(RHO*1.11) $ IOK=3H NO
      STLT1=ALAT(IU1) $ STLN1=ALON(IU1)
      STLT2=ALAT(IU2) $ STLN2=ALON(IU2)
      STLT3=ALAT(IU3) $ STLN3=ALON(IU3)
       AVLAT = ( 0.333 *(STLT1 + STLT2 + STLT3))/57.2958
       AVLON = ( 0.333 *(STLN1 + STLN2 + STLN3))/57.2958
       FC = 14.584 * SIN(AVLAT) $ COSLAT=COS(AVLAT)
       DENOM = (STLT2 -STLT1) * (STLN3 - STLN1) - (STLT3 -STLT1) *
     +      (STLN2 - STLN1)
 100  CALL RECORD
      IF (IEOF.EQ.3HYES) GO TO 200
C
C          PROCESS ONE RECORD OF DATA.
C
      DO 175 I=1,80,8
C
C          CHECK FOR STATION OF INTEREST.
C
      ISITE=INTXX(ID(I),1,5)
      DO 105 J=1,NSTA
      IF (ISITE.NE.ISTA(J)) GO TO 105
      JS=J $ GO TO 110
 105  CONTINUE
      GO TO 175
C
C          PROCESS DATA FOR ONE STATION.
C
 110  IYR=INTXX(ID(I),6,2) $ IMO=INTXX(ID(I),8,2)
      IDA=INTXX(ID(I),10,2) $ IHOUR=INTXX(ID(I+1),2,2)
      DAY=30.5*(IMO-1)+IDA
      IGH=IHOUR+IGMT(JS)
C
C          CHANGE THE DATES TO BE CONSISTENT WITH THE GMT TIMES.
C
      IF (IGH.LE.24) GO TO 111
      IGH=IGH-24 $ IDA=IDA+1
      IF (IDA.LE.MODA(IMO)) GO TO 111
      IDA=1 $ IMO=IMO+1
      IF (IMO.LE.12) GO TO 111
      IMO=1 $ IYR=IYR+1
  11  IDATE=IYR*10000+IMO*100+IDA
C
C          SKIP NON-3-HOURLY OBSERVATION.
C
      IF (MOD(IGH,3).NE.0) GO TO 175
      IS=IS+1
      IF (IGH.NE.IGHS) GO TO 180
 112  STAN(JS)=ISITE $ IOK=3H NO
      IDATES=IDATE $ IGHS=IGH
      DECODE (2,12,ID(I+4)) IWS
      IF (IWS.EQ.2H  ) GO TO 115
      WS(JS)=INTXX(ID(I+4),1,2)
      GO TO 120
 115  WS(JS)=WSS(JS) $ WRITE (6,16) ISITE,IDATE,IGH
 120  DECODE (10,13,ID(I+3)) IWD
      IF (IWD.EQ.2H  ) GO TO 125
```

```
      WD(JS)=INTXX(ID(I+3),9,2)*10.0
      GO TO 130
  125 WD(JS)=WDS(JS) $ WRITE (6,17) ISITE,IDATE,IGH
  130 DECODE (5,14,ID(I+3)) ISP
      IF (ISP.EQ.4H    ) GO TO 135
      SP(JS)=INTXX(ID(I+3),2,4)*0.1
      IF (SP(JS).LT.500.0) SP(JS)=SP(JS)+1000.0
      GO TO 140
  135 SP(JS)=SPS(JS) $ WRITE (6,18) ISITE,IDATE,IGH
  140 DECODE (9,15,ID(I+7)) IOC
      IF (IOC.EQ.1H ) GO TO 145
      IF (IOC.EQ.1HX) GO TO 150
      OC(JS)=INTXX(ID(I+7),9,1)
      GO TO 155
  145 OC(JS)=OCS(JS) $ WRITE (6,19) ISITE,IDATE,IGH
      GO TO 155
  150 OC(JS)=10.0
C
C          CHECK TO SEE IF ALL SITES ARE ACCOUNTED FOR.
C
  155 IF (IS.NE.NSTA) GO TO 175
C
C          CALCULATE THE STABILITY INDICES.
C
      DO 160 J=1,NSTA
      SP180=P180S(J) $ CP180=P180C(J)
      CALL STABLE (IHOUR,SAL,SI(J),WS(J),OC(J))
C
C          CONVERT WIND SPEED UNITS FROM KTS TO M/S.
C
      WS(J)=WS(J)*0.5148
      ANGLE=ACR*WD(J) $ U(J)=-WS(J)*SIN(ANGLE) $ V(J)=-WS(J)*COS(ANGLE)
  160 CONTINUE
C
C          CALCULATE THE GEOSTROPHIC WINDS.
C
      CALL GWINDS (UG,VG)
      WRITE (2) IDATE,IGH  ,NSTA,(STAN(L),SP(L),WD(L),WS(L),OC(L),U(L),
     1V(L),SI(L),L=1,NSTA),UG,VG
      NRW=NRW+1
C     IF (IDA.NE.15) GO TO 165
      IF (IDA.NE.15) GO TO 165
      WRITE(6,IFMT5) IDATE,IGH,(STAN(L),U(L),V(L),SI(L),L=1,NSTA),UG,VG
      WRITE (6,9) (SP(L),L=1,NSTA)
C
C          SAVE THE 3-HOURLY OBSERVATIONS.
C
  165 DO 170 J=1,NSTA
      SPS(J)=SP(J) $ WDS(J)=WD(J) $ WSS(J)=WS(J) $ OCS(J)=OC(J)
      STANS(J)=STAN(J) $ US(J)=U(J) $ VS(J)=V(J) $ SIS(J)=SI(J)
  170 CONTINUE
      UGS=UG $ VGS=VG
      IS=0 $ IOK=3HYES
  175 CONTINUE
      GO TO 100
  180 IS=1
      IF (IOK.EQ.3HYES) GO TO 112
      WRITE (2) IDATES,IGHS,NSTA,(STANS(L),SPS(L),WDS(L),WSS(L),OCS(L),
     1US(L),VS(L),SIS(L),L=1,NSTA),UGS,VGS
      NRW=NRW+1
      IF (IDA.NE.15) GO TO 112
```

```
      WRITE(6,IFMT5) IDATES,IGHS,(STANS(L),US(L),VS(L),SIS(L),L=1,NSTA),
     1UGS,VGS
      GO TO 112
  200 WRITE(6,IFMT5)IDATE,IGH  ,(STAN(L),U(L),V(L),SI(L),L=1,NSTA),UG,
     1VG
      END FILE 2 $ WRITE (6,8) NRW
      STOP200
      END
      SUBROUTINE GWINDS (UGS,VGS)
      COMMON /DAT/ PRS(10),STLAT(10),STLON(10),IU1,IU2,IU3,C1,C2,FC,
     1AVLAT,AVLON,DENOM,STLT1,STLT2,STLT3,STLN1,STLN2,STLN3,COSLAT
        PR1 = PRS(IU1)*C1
        PR2 = PRS(IU2)*C1
        PR3 = PRS(IU3)*C1
C* CORIOLIS FORCE IN UNITS 10 -5 SEC -1
C*  DENSITY IN UNITS 10 -3 G/CM3, PRESSURE IN MB
        DPDLT = (( STLN2 -STLN1) * (PR3 - PR1) - (STLN3 - STLN1) *
     +      (PR2 - PR1))/(-DENOM)
        DPDLN = (( STLT2 -STLT1) * (PR3 -PR1) - (STLT3 - STLT1) *
     +      (PR2 - PR1))/DENOM
        UGS = -(C2/FC) * DPDLT
        VGS =  (C2/FC) * (DPDLN/COSLAT)
C*  SPEED UNITS ARE  M PER SEC
      RETURN
      END
```

```
      SUBROUTINE STABLE (J,SAL,SI,WSP,OC)
C
C         THIS SUBROUTINE DETERMINES A STABILITY INDEX THROUGH A SERIES
C     OF CRITERIA CONCERNING CLOUD COVER, WIND SPEED, AND SOLAR ELEVATION
C     (SI=IJ=STABILITY INDEX, I=J=HOUR, SAL=SIN OF SOLAR ELEVATION.
C     WSP=WIND SPEED (KTS), OC=OPAQUE CLOUD COVER (TENTHS) )
C
      DIMENSION IX(15),HCOS(24)
      COMMON /CSTAB/SP180,CP180,DAY
      DATA HCOS /-0.9695,-0.866,-0.7071,-0.5,-0.2588,0.0,0.2588,0.5,
     10.7071,0.866,0.9659,1.0,0.9659,0.866,0.7071,0.5,0.2588,0.0,-0.2588
     2,-0.5,-0.7071,-0.866,-0.9659,-1.0/
      DATA IX /1,2*2,1,2,3,2,4*3,4,3,2*4/
      I=J
      CC=OC*0.1
      IJ=4
C
C         CALCULATE THE SIN OF THE SUNS ELEVATION ANGLE (SAL)
C
      XT=-.43378*COS(0.0172142*(10.0+DAY))
      XS=XT/SQRT(1.0+XT*XT)
      XC=XS/XT
      XSP=XS*SP180 $ SCP=XC*CP180
  102 HC=HCOS(I)
      SAL=XSP+HC*XCP
C
C         IS IT OVERCAST (CC.GE.0.9)
C
      IF (CC.GE.0.9) GO TO 310
C
C         IS IT NIGHT (SAL.LT.0)
C
      IF  (SAL.LT.0.0) GO TO 305
C
C         CALCULATE DAYTIME STABILITY
C
C         IS THE SUN WITHIN 15 DEGREES OF HORIZON (SAL.LT.0.26)
C
      IF (SAL.LT.0.26) GO TO 310
C-
C         RADIATION AND CLOUD AMOUNT EFFECT (XSOL=INSOLATION)
C-
      XSOL=(1.0-0.5*CC)*SAL
      IF (XSOL.GT.0.30) GO TO 120
      IRAD=3
      GO TO 200
  120 IF (XSOL.GT.0.55) GO TO 130
      IRAD=2
      GO TO 200
  130 IRAD=1
C
C         WIND SPEED EFFECT
C
  200 IF (WSP.GT.3.0) GO TO 210
      IWS=1 $ GO TO 300
  210 IF (WSP.GT.6.0) GO TO 220
      IWS=2$ GO TO 300
  220 IF(WSP.GT.10.0) GO TO 230
      IWS=3 $ GO TO 300
  230 IF (WSP.GT.12.0) GO TO 240
      IWS=4 $ GO TO 300
```

```
240   IWS=5
300   IEX=(IWS-1)*3+IRAD
      IJ=IX(IEX)
      GO TO 310
C
C          CALCULATION OF NIGHTTIME STABILITY
C
305   IF (WSP.GT.6.0) GO TO 310
      IF (CC.GE.0.5.A.WSP.GT.3.0) GO TO 310
      IJ=5
310   SI=IJ
      RETURN
      END
```

```
      SUBROUTINE RECORD
      COMMON /REC/ ID(80),NF,NR,NP,LEN,IEOF
      DATA IEOF /3H NO/
1     FORMAT (/1H ,*EOF NO. =*I3/)
2     FORMAT (1H ,*REC. NO. =*I5/)
3     FORMAT (1H ,*P.E. NO. =*I5/)
      BUFFER IN (1,0) (IDAT(1),IDAT(80))
      IF (UNIT(1)) 120,100,110
100   NF=NF+1 $ PRINT 1,NF $ IEOF=3HYES $ RETURN
110   NP=NP+1 $ PRINT 3,NP
120   NR=NR+1
      LEN=LENGTH(1)
      RETURN
      END
```

```
          IDENT     CHAR
*
*  CHARACTER STRING TRANSFER ROUTINE
*     USE FROM FORTRAN (FTN COMPILER ONLY)
*     CALL CHAR(SOURCE, I, DESTIN, J, N)
*  TRANSFER A STRING OF -N- CHARACTERS, STARTING WITH THE I-TH CHARACTER
*  OF -SOURCE- TO CHARACTER POSITIONS J,J+1,...,J+N-1 OF -DESTIN-.
*     NOTE THAT THE ARRAYS -SOURCE- AND -DESTIN- MAY BE THOUGHT OF AS
*  CHARACTER STRINGS OF ARBITRARY LENGTH, 6 BITS PER CHARACTER, 10
*  CHARACTERS PER WORD.  THUS THE 11-TH CHARACTER OF THE STRING IS
*  ACTUALLY THE 1-ST CHARACTER OF WORD 2 OF THE ARRAY, ETC.
*
         ENTRY    CHAR
CHAR     DATA     0
         SA2      A1+4
         SA2      X2
         SB1      X2
         EQ       B1,B0,CHAR
         SA4      =1RA
         SA5      =1.0E+1P0
         NX3      B0,X5
         SA2      A1+1
         SA2      X2
         IX2      X2-X4
         PX2      B0,X2
         FX1      X2/X3
         UX1      B7,X1
         LX1      B7,X1
         SB3      X1
         PX1      B0,X1
         DX1      X1*X5
         FX1      X2-X1
         SA4      =6.0P0
         DX1      X1*X4
         SB4      X1
         SA4      =1RA
         SA2      A1+3
         SA2      X2
         IX2      X2-X4
         PX2      B0,X2
         FX1      X2/X3
         UX1      B7,X1
         LX1      B7,X1
         SB5      X1
         PX1      B0,X1
         DX1      X1*X5
         FX1      X2-X1
         SA4      =6.0P0
         DX1      X1*X4
         SB6      X1
         SA2      A1
         SA4      X2+B3
         SA3      A1+2
         SA1      X3+B5
         BX7      X1
         LX4      B4,X4
         LX7      B6,X7
         MX5      6
         SB2      B0+60
TRAS     BX6      X4*X5
         BX7      -X5*X7
```

```
        BX7     X6-X7
        SB1     B1-1
        NE      B1,B0,SHIFT
        SB6     B2-B6
        LX7     B6,X7
        SA7     X3+B5
        JP      CHAR
SHIFT   LX4     6
        LX7     6
        SB4     B4+6
        SB6     B6+6
        EQ      B4,B2,IWD
OWD     NE      B6,B2,TRAS
        SA7     X3+B5
        SB5     B5+1
        SA1     X3+B5
        BX7     X1
        SB6     B0
        JP      TRAS
IWD     SB3     B3+1
        SA4     X2+B3
        SB4     B0
        JP      OWD
         END
```

```
      PROGRAM XFORM (INPUT, OUTPUT, TAPE16, TAPE3, PUNCH)
C
C     THIS PROGRAM CALLS OTHER ROUTINES FOR MAJOR OPERATIONS
      DIMENSION X(1000,20),XM(20),NBR(6),TEMP(20),A(20)
     1,D(20),Z(20,20),WK(20)
C
C     NVAR  =  NUMBER OF VARIABLES (TWICE THE NUMBER OF WINDS
C              (INCLUDING GEOSTROPHIC))
C     NREC  =  NUMBER OF RECORDS (HOURS OF DATA)
C    NSITE  =  NUMBER OF SITES (NVAR/2)-1
      READ 31,NRECS
 31   FORMAT(3I4)
      PRINT 32, NRECS
 32   FORMAT(1X,8H NRECS= ,I4)
      READ(16) IDATE,IHR,NSITE
      REWIND 16
      NVAR=2*(NSITE+1)
      PRINT 1601, NVAR,NSITE
 1601 FORMAT(1X,7H NVAR= ,I6,8H NSITE= ,I6)
C
C     SETTING CONTROL PARAMETER FOR MATRIX OPERATING ROUTINES IN SUB1
      NBR(1)=NVAR
      NBR(2)=NRECS
      NBR(3)=732
      NBR(4)=1
      NBR(5)=1
      NBR(6)=0
      IX=732
C
C     SUB1 READS INPUT DATA, CALCULATES EIGENVECTORS, INNER PRODUCTS, ETC.
      CALL SUB1(X,XM,NBR,NVAR,NSITE,NRECS,IX,A,D,Z,WK,TEMP)
      STOP
      END
```

```
      SUBROUTINE SUB1(X,XM,NBR,NVAR,NSITE,NRECS,IX,A,D,Z,WK,TEMP)
      DIMENSION X(732,NVAR),XM(NVAR),NBR(6),TEMP(NVAR),
     1VCV(100),A(20),D(NVAR),Z(NVAR,NVAR),WK(20)
      DIMENSION SUMU(10),SUMV(10),ISTAB(10)
C
C     THIS SUBROUTINE READS INPUT WIND AND STABILITY DATA, CALCULATES
C     COVARIANCE MATRIX, OBTAINS EIGENVECTORS OF COVARIANCE MATRIX, AND
C     CALLS A ROUTINE TO OBTAIN TRANSFORMED DATA.
C
C     MATSUB=NO. OF SUBMATRICES READ--GENERAL 4/YEAR
      MATSUB=4
      MVP=NSITE+1
      NCASES=0
      IPRINT=0
      DO 150 NN=1,MATSUB
C
C     AT 8 OBS/DAY THERE ARE 2928 OBS/LEAP YEAR. 2928/4=732
      DO 100 L=1,732
      READ(16) IDATE,IHR,NSITE,(DUM1,DUM2,DUM3,DUM4,DUM5,
     1SUMU(J),SUMV(J),ISTAB(J),J=1,NSITE),UG,VG
      IF(L.LE.10)
     1PRINT 1600, IDATE,IHR,NSITE,(SUMU(J),SUMV(J),ISTAB(J),J=1,92),
     1UG,VG
 1600 FORMAT(1H0,I6,I2,I3/(3F10.2))
C
C     READING INPUT DATA
C         IDATE = DATA (YR/MO/DA)
C         IHR   = HOUR OF DAY (LST)
C         SUMU,SUMV = U,V COMPONENTS OF WIND (M/S)
C         ISTAB = PASQUILL/GIFFORD STABILITY
C         UG,VG = GEOSTROPHIC WIND COMPONENTS (M/S)
      IF (EOF(16).NE.0)GOTO 99
      IF(NCASES .GT. NRECS)GOTO 99
      NCASES=NCASES+1
      SUMU(MVP)=UG
      SUMV(MVP)=VG
      DO 100 I=1,MVP
C
C     ENTERING WIND DATA INPUTS IN MATRIX X
      NPI=2*(I-1)+1
      NPIP1=NPI+1
      X(L,NPI)=SUMU(I)
      X(L,NPIP1)=SUMV(I)
C     PRINT 2003, L,NPI,NPIP1,I,X(L,NPI),X(L,NPIP1)
C2003 FORMAT(1X,*L=*,I4,* NPI=*,I4,* NPIP1=*,I4,*   I=*,I4,*  X=*,
C    12F10.2)
      IPRINT=IPRINT+1
 100  CONTINUE
 99   CONTINUE
      CALL BECOVM(X,IX,NBR,TEMP,XM,VCV,IER)
C     ******************************************************************
C
C     THE FOLLOWING DESCRIPTION OF SUBROUTINE BEVCOVM(X,IX,NBR,TEMP,XM,VC
C     V,IER) IS FROM THE IMSL MANUAL.
C
C     CALCULATES MEANS AND VARIANCE/COVARIANCE MATRIX
C     X---ON INPUT X IS A NBR(3) BY NBR(1) SUBMATRIX OF THE MATRIX (CALL
C     XX) OF DATA FOR WHICH MEANS, VARIANCES AND COVARIANCES, OR CORREC
C     TED SUMS OF SQUARES AND CROSS-PRODUCTS ARE DESIRED. THE LAST SUBMATR
C     IX IN XX MAY HAVE FEWER THAN NBR(3) ROWS.
C     ON  OUTPUT, THE ROWS OF X HAVE BEEN ADJUSTED BY THE TEMPORARY MEANS
```

```
C     IX---ROW DIMENSION OF X EXACTLY AS DIMENSIONED IN THE CALLING PROGRAM.
C     NBR---INPUT VECTOR OF LENGTH 6. NBR(I) CONTAINS, WHEN
C          I=1, NUMBER OF VARIABLES
C          I=2, NUMBER OF OBSERVATIONS PER VARIABLE IN XX
C          I=3, NUMBER OF OBSERVATIONS PER VARIABLE IN EACH SUBMATRIX X,
C               NOT INCLUDING THE LAST SUBMATRIX WHERE THE NUMBER MAY BE
C               LESS THAN OR EQUAL TO NBR(3). HOWEVER, NBR(3) SHOULD BE
C               THE SAME FOR ALL CALLS.
C          I=4, THE NUMBER OF THE SUBMATRIX STORED IN X.
C          I=5, THE TEMPORARY MEAN INDICATOR. IF NBR(5)=0, THE USER SUPPL
C               LIES  TEMPORARY MEANS IN TEMP.  OTHERWISE, THE 1ST ROW OF
C               XX (OR FIRST OF X WHEM NBR(4)=1) IS USED.
C          I=6, THE VCV OPTION. IF NBR(6)=0, VCV CONTAINS THE VARIANCE-
C               COVARIANCE MATRIX. OTHERWISE VCV CONTAINS THE CORRECTED
C               SUMS OF SQUARES AND CROSS-PRODUCTS MATRIX.
C     TEMP---INPUT VECTOR OF LENGTH NBR(1). IF NBR(5)=0 TEMP MUST CONTAIN
C          THE  TEMPORARY MEANS WHEN NBR(4)=1 OTHERWISE TEMP IS WORK STOR
C          AGE.
C     XM---OUTPUT VECTOR OF LENGTH NBR(1) CONTAINING THE VARIABLE MEANS.
C     VCV---OUTPUT NBR(1) BY NBR(1) MATRIX STORED IN SYMMETRIC STORAGE
C           MODE REQUIRING (NBR(1)*NBR(1)+1))/2 STORAGE LOCATIONS. VCV CONT
C          AINS THE VARIANCE/COVARIANCE MATRIX OR THE CORRECTED SUM OF SQ-
C          UARES AND CROSS PRODUCTS MATRIX, AS CONTROLLED BY VCV OPTION,
C          NBR(6).
C     IER---ERROR PARAMETER,  TERMINAL ERROR =128+N.  N=1 INDICATES THAT
C          NBR(4) IS LESS THAN 1 OR THAT NBR(3)*(NBR(4)-1) EXCEEDS NBR(2)
C          N=2 INDICATES THAT NBR(1) IS LESS THAN 1 OR NBR(2) IS LESS THA
C              N 2 OR THAT NBR(3) EXCEEDS NBR(2).
C     *********************************************************************$$$$$$
C
      NBR(4)=NN+1
 150  CONTINUE
C
C     PRINTING MEANS AND NUMBER OF INPUT DATA SETS
      PRINT 12,(XM(I),I=1,NVAR)
 12   FORMAT(1H0,16HVECTOR OF MEANS ,8E12.5)
      PRINT 88,NCASES
 88    FORMAT(1H0,16HNUMBER OF CASES ,I5)
C
C     OBTAINING (FROM SYMMETRIC STORAGE) AND PRINTING THE VARIANCE/COVARIANCE
C     MATRIX (VCV) AND THE ERROR PARAMETER
      DO 10 I=1,NVAR
      DO 1000 K=1,NVAR
 1000 A(K)=0.
      DO 111 J=1,NV^R
      IF (J .GT. I)K=(J*(J-1)/2)+I
      IF (J .LE. I)K=(I*(I-1)/2)+J
 111  A(J)=VCV(K)
 10   PRINT 14,(A(K),K=1,NVAR)
 14   FORMAT(1H0,18HCOVARIANCE MATRIX ,8E12.5)
      PRINT 15,IER
 15   FORMAT(1H0,21HERROR PARAMETER IS = ,I5)
C     COVARIANCE MATRIX HAS BEEN WRITTEN, NOW DO EIGENVECTORS
      N=IZ=NVAR
      IJOB=1
      CALL EIGRS(VCV,N,IJOB,D,Z,IZ,WK,IER)
C     ****************************************************************
C
C     THE FOLLOWING DESCRIPTION OF SUBROUTINE EIGRS(A,N,IJOB,D,Z,IZ,WK,IE
C     R) WAS EXTRACTED FROM THE IMSL MANUAL.
C
```

```
C     IT CALCULATES EIGENVALUES AND EIGENVECTORS
C     OF A REAL SYMMETRIC MATRIX.
C     VCV---THE INPUT SYMMETRIC MATRIX OF ORDER N, STORED IN SYMMETRIC STOR
C          AGE MODE (OBTAINED FROM BEVCOVM).
C     N---ORDER OF INPUT MATRIX VCV.
C     IJOB---INPUT OPTION PARAMETER, WHEN
C     IJOB=0, COMPUTE EIGENVALUES ONLY.  IJOB=1--EIGENVALUES AND EIGEN V
C     ECTORS.  IJOB=2--E-VALUES,E-VECTS AND PERFORMANCE INDEX.  IJOB=3--
C     PERFORM INDEX ONLY. PERFORM INDX RETURNED IN WK(1)--LT 1 = WELL, 1
C     TO 100 = SATISFACT., GT 100 = POORLY.
C     D---N-DIMENSIONAL VECTOR OF E-VALUES.
C     Z---N BY N MATRIX OF E-VECTORS OF VCV. E-VECTOR IN COLUMN J CORRESPO
C     NDS TO E-VALUE J, D(J).
C     IZ---ROW DIMEN. OF Z IN CALLING PROGRAM. IZ MUST BE GE 0.
C     WK---WORK AREA. LENGTH DEPENDS ON IJOB.  IJOB=1 OR 2, LENGTH GE N.
C     IJOB=2, LENGTH GE N(N+1)/2+N
C     IER---ERROR PARAMETER.  TERMINAL ERROR   IER=128+J, INDICATES FAILU
C     RE TO CONVERGE ON EIGENVALUE J. E-VALUES AND E-VECTORS TO J-1 ARE C
C     ORRECT, BUT E-VALUES ARE UNORDERED.
C
C     *************************************************************************
C
C      PRINTING AND PUNCHING EIGENVECTORS
       DO 16 I=1,N
 16    PRINT 32,(Z(I,J),J=1,N)
 32    FORMAT(1H0,23HMATRIX OF EIGENVECTORS ,8E12.5/26X,8E12.5)
       DO 310 I=1,N
       PUNCH 3001, I
       PRINT 3001, I
C
C      PRINTING AND PUNCHING EIGENVECTORS (X100 TO CONVERT TO CM/SEC)
 3001  FORMAT (6X,I2)
       DO 310 J=2, N, 2
       JMONE = J-1
       UCMPS = 100. * Z(JMONE,I)
       VCMPS = 100. * Z(J,I)
       PUNCH 3002, UCMPS, VCMPS
       PRINT 3002, UCMPS, VCMPS
 3002  FORMAT (2F10.2)
  310  CONTINUE
       I = N+1
       PUNCH 3001, I
       PRINT 3001, I
       DO 320  J=2,N,2
C      PRINTING AND PUNCHING MEAN VECTORS (X100 TO CONVERT TO CM/SEC)
       JMONE = J-1
       UCMEAN = 100.*XM( JMONE ).
       VCMEAN = 100.*XM( J )
       PUNCH 3002, UCMEAN, VCMEAN
       PRINT 3002, UCMEAN, VCMEAN
  320  CONTINUE
C
C      PRINT EIGENVALUES
       PRINT 112,(D(K),K=1,NVAR)
 112   FORMAT(1H0,16HEIGENVALUES ARE ,8E12.5)
       PRINT 15,IER
C      EIGENVECTOR MATRIX HAS BEEN WRITTEN, NOW DO INNER PRODUCTS
       REWIND 16
C
C      SUB2 CALCULATES INNER PRODUCTS OF INPUT DATA SETS WITH EIGENVECTORS.
       CALL SUB2(Z,NVAR,NSITE,XM,NRECS)

       RETURN
       END
```

```
      SUBROUTINE SUB2(Z,NVAR,NSITE,XM,NRECS)
C
C         Z = MATRIX OF EIGENVECTORS OF DATA COVARIANCE MATRIX
C      NVAR = NUMBER OF ELEMENTS IN INPUT DATA SETS (WIND COMPONENTS)
C        XM = MEANS OF INPUT DATA
C     NRECS = NUMBER OF INPUT DATA SETS
C     NSITE = NUMBER OF SITES USED
C     SUMU,SUMV = OBSERVED WIND COMPONENTS
      DIMENSION ISTAB(10),Z(NVAR,NVAR),XM(NVAR),
     1C(20),SUMU(10),SUMV(10),XD(20)
      DATA C/20*(-999.)/
      NCASES=0
      DO 150 L=1,NRECS
C
C     READING INPUT DATA--DATE, HOUR (LOCAL TIME) AND NSITE GROUPS
C     OF THREE (U,V,STABILITY CLASS), FINAL PAIR OF DATA ARE GEOSTROPHIC
C     U AND V.
      READ(16) IDATE,IHR,NSITE,(DUM1,DUM2,DUM3,DUM4,DUM5,
     1SUMU(J),SUMV(J),ISTAB(J),J=1,NSITE),UG,VG
      IF (EOF(16).NE.0)GOTO 99
      NSP1 = NSITE+1
      SUMU(NSP1)=UG
      SUMV(NSP1)=VG
      NCASES=NCASES+1
      DO 100 I=1,NSP1
      J=2*(I-1)+1
      JP1=J+1
C
C     GETTING DEVIATION FROM MEAN FOR EACH INPUT WIND DATUM
      XD(J)=SUMU(I)-XM(J)
      XD(JP1)=SUMV(I)-XM(JP1)
  100 CONTINUE
C     COMPUTE INNER PRODUCTS.  C(K) = INNER PRODUCT OF INPUT WIND DATA
C     (DEVIATIONS FROM MEAN) SET WITH KTH EIGENVECTOR
      DO 120 K=1,NVAR
      TERM=0.
      DO 130 J=1,NVAR
      TERM=TERM+XD(J)*Z(J,K)
  130 CONTINUE
      C(K)=TERM
  120 CONTINUE
      CONTINUE
C     WRITE DATE,HOUR,INNER PRODUCTS AND STABILITIES
      WRITE(3) IDATE,IHR,NVAR,NSITE,(C(J),J=1,NVAR),
     1(ISTAB(J),J=1,NSITE)
      IF(L.LT.30) PRINT 75,(C(LF),LF=1,NVAR)
  75  FORMAT(1X,10F10.2)
  150 CONTINUE
      PRINT 18,NCASES
  18  FORMAT(1H0,22HNUMBER OF DAYS READ = ,I5)
   99 END FILE 3
      REWIND 3
      RETURN
      END
/EOR
3636
```

```
      PROGRAM CMPLX3
C
C**THIS FORM OF PROGRAM COMPLEX COMPUTES TOPOGRAPHICALLY INDUCED
C* WINDS BY MAKING THE ORIGINALLY ANALYZED WINDS NONDIVERGENT.
C* -----FEB '85
C  THIS FORM REPLACES INWND WITH WXANAL WHICH READS IN WIND
C  SOUNDINGS AND NWS DATA  AND MAKES THE INITIAL WIND ANALYSIS.
C* THIS FORM USES DIRECT VECTOR ALTERATIONS IN SUBROUTINE BAL5.
C  NEW FEATURES INTRODUCED IN '84 - '85 ARE:
C  FOR HIGH TERRAIN SOME SIGMA SURFACES INTERSECT THE GROUND,
C  THIS IS CONTROLLED BY TERLIM IN SUBR TOPO.  ALSO,
C  WIND COMPONENTS AT SPECIFIED POINTS CAN BE HELD CONSTANT IN BAL5.
C  BY R. ENDLICH,    F. LUDWIG, K. NITZ, C. MAXWELL,  SRI INTERNAT.
C   333 RAVENSWOOD AVE MENLO PARK CALIF 94025 PHONE 415-859-3395
C
      DIMENSION B(50,26),LPRNT(10),IB(50,26)
      COMMON/RARS/RHS(50,26,6)
      COMMON/CSFC/SFCHT(50,26),SIGMA(6)
      COMMON/UARS/U(50,26,6),UA(50,26,6),V(50,26,6),VA(50,26,6)
      COMMON/WARS/W(50,26,6),WA(50,26,6)
      COMMON/PARMS/ZTOP,DS,DSIGMA,NLVLM1,XHT1,XHT2,X1,Y1,
     1 X2,Y2,UG,VG,RATIO,TDSI
      COMMON /CVOS/ RCM,RMF,IV,DSCRS,IXCRS,JYCRS,IXMED,JYMED,
     + IXFIN,JYFIN
      COMMON/CTOP/ MCRS,NCRS,MMED,NMED,MFIN,NFIN,NGRID
      COMMON/LIMITS/NCOL,NROW,NLVL,NCOLM1,NROWM1
      COMMON /BLHT/ BLT(50,26),HSITE, AVTHK, SLFAC,STHK,BLGRX,BLGRY
      COMMON /SITE/ IXS, JYS, THSITE, IGRID
      COMMON /ANCHOR/ SLAT, SLNG
      COMMON /GROUND/ TERLIM
      COMMON /PRLIM/ I1, I2
      INTEGER DNI
C  SET LIMIT ON NO. OF ITERATIONS IN SUBR. BAL
      DATA NIT/15/
C  SET NUMBER OF VERTICAL LEVELS
      DATA NLVL/6/
C  SET SIGMA LEVELS (VERTICAL COORDINATES)
      DATA SIGMA/.00, .10, .25, .45, .70, 1.00/
C  SET NO. OF GRIDS TO BE USED (1 FOR SANTA BARBARA)
      DATA NGRID/1/
C  SET NO. COLS,ROWS FOR COARSE,MED,FINE GRIDS
      DATA MCRS/50/,NCRS/26/,MMED/24/,NMED/23/
      DATA MFIN/24/,NFIN/24/
C!SET GRID INTERVALS IN KM
      DATA DSCRS/4.0/, DSMED/2.0/, DSFIN/1.0/
C SET RATIOS OF GRID INTERVALS, COARSE TO MED. , MED. TO FINE
      DATA RCM/2.0/, RMF/2.0/
C  SET HEIGHT (METERS) OF ANCHOR POINT (REFERENCE POINT)
      DATA HSITE/  0.0/
C  SET LATITUDE, LONGITUDE OF ANCHOR POINT
     DATA SLAT/33.91/, SLNG/-120.63/
C  SET ANCHOR POINT LOCATION (IX,JY) IN EACH GRID
      DATA IXCRS/ 1/,JYCRS/ 1/,IXMED/10/,JYMED/ 7/
      DATA IXFIN/12/,JYFIN/11/
C  SET LIMITS FOR PRINTING COLUMNS
      DATA I1/9/, I2/41/
C*  IN ALL ARRAYS POINT (1,1) IS AT SW CORNER.  X INCREASES TO EAST, Y
C    TO N. INDICES ARE I,J,K -(COL,ROW,LYR) WITH LIMITS NCOL,NROW,NLVL
C  UNITS USED IN COMPUTATIONS ARE M, G, SECONDS
    1 FORMAT(4I5,F10.2)
    2 FORMAT   (/'  FINAL RESULTS AT ANCHOR PT'/2X,'K       UA       VA
```

```
     +        REL. HTS.'/)
   3   FORMAT(1X,I2,2X,2F10.2,F11.3,F10.1)
   4   FORMAT(10X,'TOTAL TIME =',E12.3)
   5   FORMAT (2X,I10,2X,3F10.2,F11.3,F10.1)
   7   FORMAT(2X,5E12.3)
   8   FORMAT(6I5,2F10.2)
   9   FORMAT (2I5,2F10.2,I5,F10.0,F5.1,F8.0)
  10   FORMAT (1X,I2,2X,2F10.2,F10.1)
       PRINT 9013, MCRS,NCRS,MMED,NMED
       PRINT 9014, DSCRS, DSMED, DSFIN
C  READ BOUNDARY LAYER VARIABLES (USED IN SETBLT)
       READ(2,9022) AVTHK, SLFAC, STHK, DNI, BLGRX, BLGRY
       PRINT 9025, AVTHK, SLFAC, STHK, DNI, BLGRX, BLGRY
C  UNITS ARE METERS
C  READ TERRAIN INTERSECTION (USED IN TOPO), DATE, HOUR
       READ (2,9023) TERLIM, IDATE, IHOUR
       PRINT 9026, TERLIM
       PRINT 9027, IDATE, IHOUR
C* READ PRINT INDICATORS. TO PRINT LEVEL K USE LPRNT(K) = 1.
       READ(2,9030) (LPRNT(K),K=1,NLVL)
       PRINT 9035
       PRINT 9030, (LPRNT(K),K=1,NLVL)
C  LOOP THRU NGRID SYSTEMS.
       DO 1040 IGRID = 1,NGRID
        IF (IGRID .EQ. 2) PRINT 9005
        IF (IGRID .EQ. 3) PRINT 9006
C   SET CONTROLS FOR PROPER GRID
       IF (IGRID .NE. 1) GO TO 11
       IXS= IXCRS
       JYS= JYCRS
       NCOL= MCRS
       NROW= NCRS
       DS= DSCRS
  11   CONTINUE
       IF (IGRID .NE. 2) GO TO 14
       IXS= IXMED
       JYS= JYMED
       NCOL= MMED
       NROW= NMED
       DS= DSMED
 14    CONTINUE
       IF (IGRID .NE. 3) GO TO 15
       IXS= IXFIN
       JYS= JYFIN
       NCOL= MFIN
       NROW= NFIN
       DS= DSFIN
 15    CONTINUE
C  READ TERRAIN HEIGHTS FOR ALL GRIDS USING TOPO
       NUM1  = 0
       IF (IGRID .EQ. 1) CALL TOPO(NUM1)
       DS=DS*1.0E3
       NCOLM1=NCOL-1
       NROWM1=NROW-1
       NLVLM1=NVLV-1
       TDSI = 1./(2.0*DS)
C SET BL TOP AND SIGMA SURFACES
       CALL TOPO(IGRID)
C PRINT + PLOT SURFACE HEIGHT
       PRINT 171
 171   FORMAT (1H1,'    TERRAIN HEIGHT,  METERS,  NORTH ROW FIRST'/
```

```
      DO 53 JP =1,NROW
      DO 53 IP =1,NCOL
            B(IP,JP) =   SFCHT(IP,JP)
           IB(IP,JP)= JNINT(B(IP,JP))
 53   CONTINUE
      DO 54 JP =1,NROW
            JR = NROW+ 1 -JP
 54   PRINT 9105,  (   IB(IP,JR),IP=I1,I2 )
 55   CONTINUE
C * * * *PLOT GEOMETRIC HEIGHTS OF  SELECTED SIGMA SURFACES
      DO 511  K =1,NLVL
      IF (LPRNT(K) .NE. 1) GO TO 511
      DO  627  JP = 1,NROW
      DO 627 IP = 1,NCOL
            B(IP,JP) =   RHS(IP,JP,K)
            IB(IP,JP)= JNINT(B(IP,JP))
 627  CONTINUE
      PRINT 571,K
 571  FORMAT (1H1,'    HEIGHT ABOVE TERRAIN, M,  LVL='I3/)
      DO 628 JP=1,NROW
      JR = NROW+ 1 -JP
 628  PRINT 9105,  (IB(IP,JR),IP=I1,I2 )
 511  CONTINUE
C* READ AND ANALYZE WIND DATA USING WXANAL
C* MAKE INITIAL WIND ANALYSIS ON MESH
      CALL WXANAL(IGRID)
C * * * * * *PLOT  OBSERVED VELOCITY COMPONENTS AT SELECTED LEVELS * *
      DO 211  K =1,NLVL
      IF (LPRNT(K) .NE. 1) GO TO 211
      DO 46 J =1,NROW
      DO 46 I =1,NCOL
        IB(I,J) = JNINT( U(I,J,K)*10.)
 46   CONTINUE
      PRINT 271,K
 271  FORMAT(1H1,'  U COMPONENT DECIMETERS/SEC, LVL = 'I4/)
      DO 56 JP =1,NROW
      JR = NROW+ 1 -JP
 56   PRINT 9105,  (   IB(I,JR),I=I1,I2 )
 211  CONTINUE
      DO 212  K =1,NLVL
      IF (LPRNT(K) .NE. 1) GO TO 212
      DO 48 J =1,NROW
      DO 48 I =1,NCOL
        IB(I,J) = JNINT( V(I,J,K)*10.)
 48   CONTINUE
      PRINT 272,K
 272  FORMAT(1H1,'  V COMPONENT DECIMETERS/SEC, LVL = 'I4/)
      DO 58 JP =1,NROW
      JR = NROW+ 1 -JP
 58   PRINT 9105,  (   IB(I,JR),I=I1,I2 )
 212  CONTINUE
      IF (IGRID .GT. 1)  GO TO 226
C  COMPUTE VERTICAL MOTION W ALONG SIGMA SURFACES
      DO 220  K =2,NLVL
      DO 220  I =2,NCOLM1
      DO 220  J =2,NROWM1
      W(I,J,K) = 0.0
     IF (RHS(I,J,K) .LE. 0.0) GO TO 220  !  FOR TERRAIN LIMIT
      HSIGE=SFCHT(I+1,J) + RHS(I+1,J,K)
      HSIGW=SFCHT(I-1,J) + RHS(I-1,J,K)
      HSIGN=SFCHT(I,J+1) + RHS(I,J+1,K)
```

```
      HSIGS=SFCHT(I,J-1) + RHS(I,J-1,K)
      DHDX=(HSIGE - HSIGW) * TDSI
      DHDY=(HSIGN - HSIGS) * TDSI
      W(I,J,K)=U(I,J,K) * DHDX + V(I,J,K) * DHDY
  220 CONTINUE
  226 CONTINUE
      IF (IGRID .EQ. 1) PRINT 9050
      IF (IGRID .EQ. 2) PRINT 9055
      IF (IGRID .EQ. 3) PRINT 9056
      PRINT 9020
 9020 FORMAT (/'  ORIGINAL U, V, W, REL. HTS AT ANCHOR PT.'/)
      PRINT 3,(K, U(IXS,JYS,K), V(IXS,JYS,K), W(IXS,JYS,K),
     2        RHS(IXS,JYS,K),K=1,NLVL)
      DO 213  K =1,NLVL
      IF (LPRNT(K) .NE. 1) GO TO 213
      PRINT 273,K
  273 FORMAT(1H1,'  W, CM/SEC , LVL ='I3/)
      DO 60  JP = 1,NROW
      DO 60  IP = 1,NCOL
       IB(IP,JP) = JNINT(W(IP,JP,K)*100.)
   60 CONTINUE
      DO 65  JP = 1,NROW
      JR = NROW+ 1 -JP
   65 PRINT 9105,  (  IB(IP,JR),IP=I1,I2)
  213 CONTINUE
C CALL SUBROUTINE TO MAKE WINDS NONDIVERGENT
      CALL BAL5(NIT)
      DO 605 I=1,NCOL
      DO 605 J=1,NROW
      DO 605 K=1,NLVL
              UA(I,J,K)=U(I,J,K)
              VA(I,J,K)=V(I,J,K)
              WA(I,J,K)=W(I,J,K)
  605 CONTINUE
C * * * *PLOT ADJUSTED VELOCITY COMPONENTS  AT SELECTED LEVELS
      DO 611  K =1,NLVL
      IF (LPRNT(K) .NE. 1) GO TO 611
      DO 615 J = 1,NROW
      DO 615 I = 1,NCOL
        IB(I,J) = JNINT(UA(I,J,K)*10.)
  615 CONTINUE
      PRINT 671,K
  671 FORMAT (1H1,'   ADJUSTED U COMPONENT, DECIMETERS/SEC,  LVL='I3/)
      DO 622 JP=1,NROW
      JR = NROW+ 1 -JP
  622 PRINT 9105,  (  IB(I,JR),I=I1,I2 )
  611 CONTINUE
      DO 612  K =1,NLVL
      IF (LPRNT(K) .NE. 1) GO TO 612
      DO 616 J = 1,NROW
      DO 616 I = 1,NCOL
        IB(I,J) = JNINT(VA(I,J,K)*10.)
  616 CONTINUE
      PRINT 672,K
  672 FORMAT (1H1,'   ADJUSTED V COMPONENT, DECIMETERS/SEC,  LVL='I3/)
      DO 624 JP=1,NROW
      JR = NROW+ 1 -JP
  624 PRINT 9105,  (  IB(I,JR),I=I1,I2 )
  612 CONTINUE
      IF (IGRID .EQ. 1) PRINT 9050
      IF (IGRID .EQ. 2) PRINT 9055
```

```
      IF (IGRID .EQ. 3) PRINT 9056
      PRINT 2
      PRINT 10,(K,UA(IXS,JYS,K),VA(IXS,JYS,K),RHS(IXS,JYS,K),
     + K=1,NLVL)
      DO 700 K =2, NLVL
        DO 690 J = 1, NROW
          JR = NROW + 1 -J
690     WRITE (3,9065) (UA(I,JR,K),I=1,NCOL)
        DO 695 J = 1, NROW
          JR = NROW + 1 -J
695     WRITE (3,9065) (VA(I,JR,K),I=1,NCOL)
700   CONTINUE
1040  CONTINUE
1050  CONTINUE
9005 FORMAT (1H1,' **BEGIN COMPUTATIONS FOR MEDIUM GRID**'/)
9006 FORMAT (1H1,' **BEGIN COMPUTATIONS FOR FINE GRID**'/)
9013  FORMAT (/' COARSE GRID E-W'I5,' S-N'I5,'  MEDIUM GRID E-W'I4,'  S-N
     + 'I4/)
9014 FORMAT (/' GRID INCREMENTS IN KM, COARSE='F4.1,' MED.='F4.1,'
     + FINE='F4.1/)
9022  FORMAT (F10.1,F10.2,F10.1,I5,2F7.1)
9023 FORMAT (F10.1,2I8)
9025  FORMAT (//'   AVER. BNDY. THICKNESS IN M ='F8.1,
     + 'SLOPE FACTOR FOR BL TOP='F4.1,' MIN. THICKNESS='F7.1,
     + /' DAY1 NITE2 INDICATOR='I3,' B LYR GRADIENT TO EAST, M='F7.1
     + ' TO NORTH =' F7.1/)
9026  FORMAT (/'    TERRAIN INTERSECTION WILL OCCUR FOR HTS
     + ABOVE'F6.1,' METERS'/)
9027 FORMAT (/'   DATE ='I8,'    HOUR ='I4/)
9030  FORMAT (12I5)
9035  FORMAT (/'   INDICATORS, LPRNT(K),FOR PRINTING FIELDS'/)
9040  FORMAT (1H1,'   BEGIN COMPUTATIONSFOR FINE GRID'/)
9045  FORMAT (1H1,'    U COMP. AT LEVEL 3, CM/SEC'/)
9046 FORMAT (1H1,'    V COMP. AT LEVEL 3  '/)
9050  FORMAT (1H1,'    COARSE GRID')
9055  FORMAT (1H1,'     MEDIUM GRID')
9056 FORMAT (1H1,'   FINE GRID')
9060  FORMAT (I5,E10.1,I5)
9065 FORMAT (10F8.1)
9100  FORMAT (/5X,22F5.0)
9105 FORMAT (/1X,33I4)
      STOP
        END
      SUBROUTINE WXANAL(NUM)
C  THIS SUBROUTINE DOES THE FOLLOWING.  WHEN NUM=IGRID=1 IT
C  READS IN DATA FROM WIND PROFILES AND ASSIGNS WINDS TO
C  SIGMA LEVELS; READS  NWS STATIONS, COMPUTES THE GEOSTROPHIC
C  WIND AND ASSIGNS WINDS TO SIGMA LEVELS, COMBINES SOUNDINGS AND
C  HOURLY DATA, AND MAKES INITIAL OBJECTIVE WIND ANALYSIS USING
C  A WT FACTOR INVERSELY PROPORTIONAL TO DISTANCE SQUARED.
C  WHEN NUM .GT. 1 (IGRID=1 OR 2) IT SELECTS INITIAL WINDS FROM
C  THE NEXT COARSER GRID.
C  BY RM ENDLICH,SRI INTN'L, JAN '85
C ------------------------------
      COMMON/LIMITS/NCOL,NROW,NLVL,NCOLM1,NROWM1
        COMMON /CVOS/ RCM,RMF,IV,DSCRS,IXCRS,JYCRS,IXMED,JYMED,
     +   IXFIN,JYFIN
      COMMON/RARS/RHS(50,26,6)
      COMMON/UARS/U(50,26,6),UA(50,26,6),V(50,26,6),VA(50,26,6)
      COMMON/WARS/W(50,26,6),WA(50,26,6)
      COMMON/CSFC/SFCHT(50,26),SIGMA(6)
```

```
      COMMON/PARMS/ZTOP,DS,DSIGMA,NLVLM1,XHT1,XHT2,X1,Y1,
     1 X2,Y2,UG,VG,RATIO,TDSI
       COMMON/CTOP/ MCRS,NCRS,MMED,NMED,MFIN,NFIN,NGRID
      COMMON /SITE/ IXS, JYS, THSITE, IGRID
      COMMON /ANCHOR/ SLAT, SLNG
      COMMON /NUMOBS/ NUMDOP, NUMNWS, NUMTOT
      COMMON /UPWIND/ UTOP, VTOP
       COMMON /STALOC/ XG(50),YG(50)
       COMMON /WINDS/ USIG(50,6), VSIG(50,6)
       DIMENSION UTEMP(50,26), VTEMP(50,26)
    2 FORMAT(4X,13F4.0)
    3 FORMAT(4X,-2P13F5.0)
    4 FORMAT(I10,2F10.1)
    5 FORMAT(3F10.2)
    6 FORMAT(4X,-2P14F6.2)
    7 FORMAT(1H )
    8 FORMAT(4X,-5P3F10.2)
       PRINT 9001, NUM
 9001 FORMAT (/' BEGIN SUBROUTINE WXANAL, NUM='I3/)
        IF (NUM .GT. 1)  GO TO 200
C  READ WIND SOUNDINGS AND ASSIGN TO SIGMA LEVELS
       CALL DOPSIG
C  READ NWS REPORTS AND INTERPOLATE TO SIGMA SFCS
       READ(2,9014) NUMNWS
       PRINT 9027, NUMNWS
       IF (NUMNWS .GE. 1)  CALL GEOSIG
C  TOTAL NUMBER OF STATIONS (NUMTOT) = NUMDOP + NUMNWS
         NUMTOT = NUMDOP + NUMNWS
C  MAKE GRID POINT ANALYSIS OF DATA
       CALL GPAN
      DO 50 J = 1, NROW
      DO 50 I = 1, NCOL
      DO 40 LV = 2,NLVL
C  TO USE RHS NEGATIVE (BELOW TERRAIN) MAKE WINDS 0.
      IF (RHS(I,J,LV) .GE. 0.0) GO TO 40
         U(I,J,LV) = 0.0
         V(I,J,LV) = 0.0
   40 CONTINUE
   50 CONTINUE
      GO TO 300
  200 CONTINUE
C  SELECT WINDS FOR SMALLER GRID FROM LARGER GRID
C  ASSIGN WINDS TO MED. GRID FROM COARSE GRID
        IF (NUM .NE. 2) GO TO 216
        DO 215 K =2,NLVL
        DO 210 I =1,NCOL
        DO 210 J =1,NROW
         IC =IXCRS + JNINT(FLOAT(I-IXMED)/RCM)
         JC =JYCRS + JNINT(FLOAT(J-JYMED)/RCM)
C  FILL IN NONZERO WINDS FROM NEAREST POINTS
       IF (U(IC,JC,K) .EQ. 0. .AND. V(IC,JC,K) .EQ. 0.) IC=IC+1
       IF (U(IC,JC,K) .EQ. 0. .AND. V(IC,JC,K) .EQ. 0.) IC=IC-2
       IF (U(IC,JC,K) .EQ. 0. .AND. V(IC,JC,K) .EQ. 0.) JC=JC+1
       IF (U(IC,JC,K) .EQ. 0. .AND. V(IC,JC,K) .EQ. 0.) JC=JC-2
         UTEMP(I,J) =U(IC,JC,K)
         VTEMP(I,J) =V(IC,JC,K)
 210    CONTINUE
       DO 212 I =1,NCOL
       DO 212 J =1,NROW
        U(I,J,K) = UTEMP(I,J)
        V(I,J,K) = VTEMP(I,J)
```

```
          IF (RHS(I,J,K) .LE. 0.0) U(I,J,K) = 0.0
          IF (RHS(I,J,K) .LE. 0.0) V(I,J,K) = 0.0
  212   CONTINUE
  215   CONTINUE
  216  CONTINUE
C  SELECT FINE GRID WINDS FROM MEDIUM GRID WINDS
          IF (NUM .NE. 3) GO TO 300
          DO 230 K =2,NLVL
          DO 225 I =1,NCOL
          DO 225 J =1,NROW
           IC =IXMED + JNINT(FLOAT(I-IXFIN)/RMF)
           JC =JYMED + JNINT(FLOAT(J-JYFIN)/RMF)
C  FILL IN NONZERO WINDS FROM NEAREST POINTS
        IF (U(IC,JC,K) .EQ. 0. .AND. V(IC,JC,K) .EQ. 0.) IC=IC+1
        IF (U(IC,JC,K) .EQ. 0. .AND. V(IC,JC,K) .EQ. 0.) IC=IC-2
        IF (U(IC,JC,K) .EQ. 0. .AND. V(IC,JC,K) .EQ. 0.) JC=JC+1
        IF (U(IC,JC,K) .EQ. 0. .AND. V(IC,JC,K) .EQ. 0.) JC=JC-2
           UTEMP(I,J) =U(IC,JC,K)
           VTEMP(I,J) =V(IC,JC,K)
  225     CONTINUE
        DO 227 I =1,NCOL
        DO 227 J =1,NROW
          U(I,J,K) = UTEMP(I,J)
          V(I,J,K) = VTEMP(I,J)
          IF (RHS(I,J,K) .LE. 0.0) U(I,J,K) = 0.0
          IF (RHS(I,J,K) .LE. 0.0) V(I,J,K) = 0.0
  227   CONTINUE
  230   CONTINUE
   300 CONTINUE
          PRINT 9002
  9002 FORMAT (/' END OF SUBROUTINE WXANAL'/)
  9007 FORMAT (//'  THE ANCHOR PT. IS AT LAT ='F9.3,'  AND LONG='F9.3)
  9014  FORMAT (3X,I5,2F8.2)
  9027  FORMAT (/'  NUMBER OF HOURLY SURFACE REPORTS ='I3/)
          RETURN
           END
        SUBROUTINE DOPSIG
C  ASSIGN DOPPLER WIND PROFILES TO SIGMA SURFACES.
C  MISSING WINDS ARE DENOTED BY -999.
C  IF SOUNDING IS NOT COMPLETE THE LAST REPORTED WIND
C  IS USED AT THE HIGHEST ALTITUDES.
C  WIND DATA START AT 40 M AND CONTINUE AT 30-M INTERVALS
C  TO 610 M.  THERE ARE 20 POINTS IN A PROFILE.
C  FILL IN MISSING DATA WITH NEAREST POINT UP OR DOWN.
C  FOR PROGRAM DIABWND, JAN  '85.
C  BY R.M. ENDLICH, SRI INTN'L, MENLO PARK CA 94025.
        DIMENSION  DPHT(5,50), DPUC(5,50), DPVC(5,50)
        DIMENSION RHS1(5,6),   STLT(5), STLN(5)
        DIMENSION DPWD(5,50),DPWS(5,50)
      COMMON /STALOC/ XG(50), YG(50)
      COMMON /WINDS/ USIG(50,6), VSIG(50,6)
        DIMENSION XS(5), YS(5)
      COMMON/LIMITS/NCOL,NROW,NLVL,NCOLM1,NROWM1
      COMMON/RARS/RHS(50,26,6)
      COMMON/UARS/U(50,26,6),UA(50,26,6),V(50,26,6),VA(50,26,6)
      COMMON/PARMS/ZTOP,DS,DSIGMA,NLVLM1,XHT1,XHT2,X1,Y1,
     1 X2,Y2,UG,VG,RATIO,TDSI
      COMMON /SITE/ IXS, JYS, THSITE, ISRID
      COMMON /ANCHOR/ SLAT, SLNG
      COMMON /NUMOBS/ NUMDOP, NUMNWS, NUMTOT
      COMMON /UPWIND/ UTOP, VTOP
```

```
      INTEGER STAID(5)
C  VARIABLES ARE:
C     DPUC = U COMPONENT OF DOPPLER WIND IN MPS
C     DPVC = V COMPONENT OF DOPPLER WIND IN MPS
C     NHTS = NUMBER OF POINTS IN VERTICAL WIND PROFILE
C     NLVL = NUMBER OF SIGMA LEVELS
C     RHS = HT OF SIGMA SURFACES ABOVE TERRAIN (M)
C     XG,YG = STA. DIST IN X,Y IN GRID UNITS FROM SW CORNER
      PRINT 9001
 9001 FORMAT (/'     BEGIN SUBROUTINE DOPSIG  '/)
C  FILL IN HEIGHTS OF DOPPLER DATA POINTS
C -----------------------------
C  PRINT LAT/LONG OF ANCHOR POINT
      PRINT 9007, SLAT, SLNG
C* READ LAT , LONG  OF STATIONS (DEG) AND CONVERT TO  XS,YS (KM)
C  MEASURED FROM SW CORNER OF COARSE GRID
      READ(2,9014) NUMDOP
      PRINT 9027, NUMDOP
      READ(2,9014) (STAID(IT),STLT(IT), STLN(IT),IT=1,NUMDOP)
      PRINT 9004
      PRINT   9014,  (STAID(IT),STLT(IT), STLN(IT),IT=1,NUMDOP)
     DO 15 IT=1,NUMDOP
      XS(IT) = (STLN(IT) - SLNG)*(111.2 *COS(SLAT/57.295))
      YS(IT) = (STLT(IT) - SLAT)*111.2
      XS(IT) = XS(IT) + IXS * (DS * .001)
      YS(IT) = YS(IT) + JYS * (DS * .001)
      XG(IT) = XS(IT)/(DS * .001)
      YG(IT) = YS(IT)/(DS * .001)
  15 CONTINUE
      PRINT 9008
      PRINT 9011, (XS(IT), YS(IT),XG(IT),YG(IT),IT=1,NUMDOP)
C  ASSIGN HTS OF SIGMA SFC FOR EACH SOUNDING
     DO 25 IT = 1,NUMDOP
     DO 20 K = 1,NLVL
      IX = JNINT(XG(IT))
      JY = JNINT(YG(IT))
      IF (IX .LT. 1) IX = 1
      IF (IX .GT. NCOL) IX = NCOL
      IF (JY .LT. 1) JY = 1
      IF (JY .GT. NROW) JY = NROW
      RHS1(IT,K) = RHS(IX,JY,K)
  20 CONTINUE
      PRINT 9024
      PRINT 9015, IT, (RHS1(IT,K),K=1,NLVL)
  25 CONTINUE
C  READ DOPPLER SOUNDINGS FOR NUMDOP STATIONS
     DO 450 IT =1,NUMDOP
       READ (2,9014) NHTS
       PRINT 9003, IT, NHTS
       READ (2,9009) ( DPHT(IT,LL),DPWD(IT,LL), DPWS(IT,LL),LL=1,NHTS)
       PRINT 9009, ( DPHT(IT,LL),DPWD(IT,LL), DPWS(IT,LL),LL=1,NHTS)
C  CHANGE DIRECTION AND SPEED (MPS) TO U AND V
       DO 40 LL = 1,NHTS
         IF (DPWD(IT,LL) .EQ. -999.0) THEN
           DPUC(IT,LL) = -999.0
           DPVC(IT,LL) = -999.0
         ELSE
           DPUC(IT,LL) = -DPWS(IT,LL) * SIN(DPWD(IT,LL)/57.295)
           DPVC(IT,LL) = -DPWS(IT,LL) * COS(DPWD(IT,LL)/57.295)
         END IF
   40    CONTINUE
```

```
      PRINT 9010, (DPUC(IT,LL), DPVC(IT,LL),LL=1,NHTS)
C  FILL IN MISSING DATA IF NEEDED
C  START FROM BOTTOM AND GO UPWARD
      LL = 0
   60 LL  = LL +1
      IF (LL .EQ. NHTS) GO TO 100
      IF (DPUC(IT,LL) .EQ. -999.0) THEN
        GO TO 70
      ELSE
        GO TO 60
      END IF
   70 LL1 = LL
   75 LL1 = LL1 + 1
      IF (DPUC(IT,LL1) .EQ. -999.0 .AND. LL1 .EQ. NHTS)
     +  GO TO 100
      IF (DPUC(IT,LL1) .EQ. -999.0) THEN
        GO TO 75
      ELSE
        DPUC(IT,LL) = DPUC(IT,LL1)
        DPVC(IT,LL) = DPVC(IT,LL1)
      PRINT 9015, LL, DPUC(IT,LL), DPVC(IT,LL)
        GO TO 60
      END IF
  100 CONTINUE
      PRINT 9010, (DPUC(IT,LL), DPVC(IT,LL),LL=1,NHTS)
C  START FROM TOP AND GO DOWN
      LL = NHTS + 1
  110 LL  = LL -1
      IF (LL .EQ. 1) GO TO 140
      IF (DPUC(IT,LL) .EQ. -999.0) THEN
        GO TO 120
      ELSE
        GO TO 110
      END IF
  120 LL1 = LL
  125 LL1 = LL1 - 1
      IF (DPUC(IT,LL1) .EQ. -999.0) THEN
        GO TO 125
      ELSE
        DPUC(IT,LL) = DPUC(IT,LL1)
        DPVC(IT,LL) = DPVC(IT,LL1)
      PRINT 9015, LL, DPUC(IT,LL), DPVC(IT,LL)
        GO TO 110
      END IF
  140 CONTINUE
      PRINT 9012
      PRINT 9010, (DPUC(IT,LL), DPVC(IT,LL),LL=1,NHTS)
C  BEGIN INTERPOLATION SCHEME
      DO 400 K = 2, NLVL  ! COUNTER FOR SIGMA LEVELS
        LL = 0
  200   LL = LL + 1
        IF (RHS1(IT,K) .GT. 0.0) GO TO 365
          USIG(IT,K) = 0.0
          VSIG(IT,K) = 0.0
          GO TO 400
  365    IF (RHS1(IT,K) .LE. DPHT(IT,LL)) GO TO 320
        IF (RHS1(IT,K) .GE. DPHT(IT,NHTS)) GO TO 380
        IF (RHS1(IT,K) .GE. DPHT(IT,LL) .AND. RHS1(IT,K) .LE.
     +   DPHT(IT,LL+1))  GO TO 360
        GO TO 200
C  FOR LEVELS BELOW 1ST MEASURED WINDS (ASSUME SPEED=0 AT 1M
```

```
C   AND AT GROUND WHERE RHS=0)
  320    USIG(IT,K)=(DPUC(IT,LL))*(ALOG10(RHS1(IT,K)))/
     +     ALOG10(DPHT(IT,LL))
         VSIG(IT,K)=(DPVC(IT,LL))*(ALOG10(RHS1(IT,K)))/
     +     ALOG10(DPHT(IT,LL))
         GO TO 400
C FOR SIGMA LEVELS BETWEEN DOPPLER WIND POINTS
  360   RATIO=(ALOG10(RHS1(IT,K))-ALOG10(DPHT(IT,LL)))/
     +     (ALOG10(DPHT(IT,LL+1))- ALOG10(DPHT(IT,LL)))
         USIG(IT,K) = DPUC(IT,LL) +(DPUC(IT,LL+1)-DPUC(IT,LL))
     +     * RATIO
         VSIG(IT,K) = DPVC(IT,LL) +(DPVC(IT,LL+1)-DPVC(IT,LL))
     +     * RATIO
         GO TO 400
C   FOR SIGMA LEVELS ABOVE LAST DOPPLER POINT
  380    USIG(IT,K) = DPUC(IT,NHTS)
         VSIG(IT,K) = DPVC(IT,NHTS)
  400   CONTINUE
        PRINT 9020, IT
        PRINT 9010, (USIG(IT,K), VSIG(IT,K), K=1,NLVL)
  450   CONTINUE
C   GET AVER UPPER WIND FOR POSSIBLE USE IN GEOSIG IF GEOS WIND
C   IS NOT AVAILABLE
           UTOP = 0
           VTOP = 0
        DO 460 IT = 1, NUMDOP
           UTOP = UTOP + USIG(IT,NLVL)
           VTOP = VTOP + VSIG(IT,NLVL)
  460   CONTINUE
           UTOP = UTOP/(FLOAT(NUMDOP))
           VTOP = VTOP/(FLOAT(NUMDOP))
        PRINT 9026, UTOP, VTOP
        PRINT 9002
 9002   FORMAT (/'         END OF SUBROUTINE DOPSIG   '/)
 9003   FORMAT (/'    STA. ='I3,'  NO. OF POINTS IN PROFILE='I3/)
 9004   FORMAT (//'   LATITUDE AND LONGITUDE OF STATIONS'/)
 9005   FORMAT (/' WIND  DIR  WIND SPEED,    STATION='I4/)
 9006   FORMAT (/' U COMPONENT V COMPONENT AT SIGMA LEVELS'/)
 9007   FORMAT (//'   THE ANCHOR PT. IS AT LAT ='F9.3,'  AND LONG='F9.3)
 9008 FORMAT (//'   X AND Y OF STATIONS IN KM AND IN COLS,ROWS FROM SW
     +   CORNER OF COARSE GRID'/)
 9009   FORMAT (3X,3F7.1)
 9010   FORMAT (3X,8F6.1)
 9011   FORMAT (/4X,2F11.0,8X,2F10.1)
 9012   FORMAT (/' SOUNDING WITH FILLED IN DATA '/)
 9014   FORMAT (3X,I5,2F8.2)
 9015   FORMAT (3X,I5,6F8.2)
 9020   FORMAT (/' STATION NUMBER ='I6/)
 9024   FORMAT (/'  HEIGHTS OF SIGMA SURFACES '/)
 9026   FORMAT (/'    AVER. UPPER WIND, U ='F6.1,'    V='F6.1/)
 9027   FORMAT (/'  NUMBER OF WIND SOUNDINGS ='I3/)
         RETURN
         END
         SUBROUTINE GEOSIG
C   PREPARE NWS HOURLY REPORTS OF WIND DIRECTION, WIND SPEED
C   (KNOTS), SEA LEVEL PRESSURE (MB), AND TEMPERATURE (DEG F)
C   FOR INPUT TO WIND ANALYSIS FOR DIABLO PGE SITE.
C   COMPUTE GEOS WIND FROM PRESSURE AT THREE STATIONS AND
C   CORRECT IT FOR THERMAL WIND COMPONENT (IF DESIRED).
C   ASSUME THAT WIND COMPONENTS VARY WITH LOG(HEIGHT) BETWEEN
C   ANEMOMETER HT AND GEOS WIND AT HT GEOSHT(ABOUT 500M).
```

```
C  AT EACH NWS STATION INTERPOLATE WINDS TO SIGMA SURFACES.
C  BY RM ENDLICH, SRI INTN'L, MENLO PARK CA 94025 DEC '84.
C  VARIABLES
C    NUMNWS = NUMBER OF NWS REPORTS
C    NWSID = IDENTIFICATION ID OF NWS STATION
C    WD = WIND DIRECTION
C    SP = WIND SPEED
C    STLT = STATION LATITUDE IN DEGS AND HUNDREDTHS
C    STLN = STATION LONGITUDE
C    PRESS = STATION SEA LEVEL PRESSURE IN INCHES HG
C    TEMP = STATION TEMP IN DEG F
        DIMENSION   STLT(50), STLN(50), PRESS(50)
        DIMENSION TEMP(50)
        DIMENSION RHS1(50,6), UNWS(50,6), VNWS(50,6)
        DIMENSION UCOMP(50), VCOMP(50), WD(50), WS(50)
        DIMENSION XS(50), YS(50)
        DIMENSION XG1(50), YG1(50), USIG1(50,6), VSIG1(50,6)
       COMMON/LIMITS/NCOL,NROW,NLVL,NCOLM1,NROWM1
       COMMON/RARS/RHS(50,26,6)
       COMMON/UARS/U(50,26,6),UA(50,26,6),V(50,26,6),VA(50,26,6)
       COMMON/PARMS/ZTOP,DS,DSIGMA,NLVLM1,XHT1,XHT2,X1,Y1,
      1 X2,Y2,UG,VG,RATIO,TDSI
       COMMON /SITE/ IXS, JYS, THSITE, IGRID
       COMMON /STALOC/ XG(50), YG(50)
       COMMON /WINDS/ USIG(50,6), VSIG(50,6)
       COMMON /ANCHOR/ SLAT, SLNG
       COMMON /NUMOBS/ NUMDOP, NUMNWS, NUMTOT
       COMMON /UPWIND/ UTOP, VTOP
        INTEGER STAID(50)
  10    FORMAT (I5,2F8.2,4F7.1)
  20    FORMAT (/' NO.='I3,' STA ID='I5,'  LAT='F7.2,'  LONG.
     + ='F7.2,'  PRESS='F7.2,'  TEMP='F6.1/)
  25    FORMAT ('     WIND DIR='F6.1,'  WIND SPEED='F5.1/)
  30    FORMAT ('   GEOSTROPHIC WIND COMPONENTS, MPS, U ='F5.1,
     + '    V ='F5.1/)
  40    FORMAT (//'    THERMAL WIND SHEAR COMPONENTS, MPS, U ='
     + F5.1,'    V ='F5.1,'  LAYER DEPTH ='F6.1,' M'/)
  50    FORMAT (//'  GEOSTROPHIC WIND CORRECTED FOR THERMAL WIND')

        PRINT 9001
 9001   FORMAT (/'      BEGIN SUBROUTINE GEOSIG  '/)
C  READ INPUT OF HOURLY DATA: STATION ID, LATITUDE, LONGITUDE,
C    PRESSURE, TEMPERATURE, WIND DIRECTION, WIND SPEED (MPS)
         DO  70 IT = 1,NUMNWS
        READ (2, 10) STAID(IT), STLT(IT), STLN(IT),
     +     PRESS(IT), TEMP(IT), WD(IT), WS(IT)
          PRINT 20, IT,  STAID(IT),STLT(IT),STLN(IT),
     +        PRESS(IT),TEMP(IT)
          PRINT 25, WD(IT), WS(IT)
   70  CONTINUE
C  CONVERT LAT,LONG TO XS,YS (KM) MEASURED FROM SW CORNER OF
C   COARSE GRID
       DO 15 J=1,NUMNWS
        XS(J) = (STLN(J) - SLNG)*(111.0 *COS(SLAT/57.295))
        YS(J) = (STLT(J) - SLAT)*111.0
        XS(J) = XS(J) + IXS * (DS * .001)
        YS(J) = YS(J) + JYS * (DS * .001)
        XG1(J) = XS(J)/(DS * .001)
        YG1(J) = YS(J)/(DS * .001)
  15   CONTINUE
        PRINT 9008
```

```
      PRINT 9011, (XS(J), YS(J),XG1(J),YG1(J),J=1,NUMNWS)
C  CHANGE DIRECTION AND SPEED (MPS) TO U AND V
       DO 80 IT = 1,NUMNWS
           UCOMP(IT) = -WS(IT) * SIN(WD(IT)/57.295)
           VCOMP(IT) = -WS(IT) * COS(WD(IT)/57.295)
       PRINT 9002
       PRINT 9013, IT, UCOMP(IT), VCOMP(IT)
   80   CONTINUE
C  ASSIGN HEIGHTS TO SIGMA SURFACES
      DO 125  J = 1,NUMNWS
      DO 120 K = 1,NLVL
        IX = JNINT (XG1(J))
        JY = JNINT (YG1(J))
        IF (IX .LT. 1) IX = 1
        IF (IX .GT. NCOL) IX = NCOL
        IF (JY .LT. 1) JY = 1
        IF (JY .GT. NROW) JY = NROW
        RHS1(J,K) = RHS(IX,JY,K)
  120 CONTINUE
      PRINT 9024
      PRINT 9015, J, (RHS1(J,K),K=1,NLVL)
  125 CONTINUE
C  FOR SANTA BARBARA NEGLECT GEOS WIND (NOT REPRESENTATIVE)
      UGEOS = UTOP
      VGEOS = VTOP
      GO TO 200
C  IF LESS THAN 3 NWS STATIONS CANT COMPUTE GEOS WIND.  INSTEAD
C  USE UTOP, VTOP FROM DOPPLER.
      IF (NUMNWS .LT. 3) UGEOS = UTOP
      IF (NUMNWS .LT. 3) VGEOS = VTOP
      IF (NUMNWS .LT. 3) GO TO 200
C**THIS SECTION COMPUTES GEOSTROPHIC WINDS AND IT ALSO CORRECTS
C  THEM FOR THERMAL WINDS IF NTHERM=1.
C  SELECT  THREE STATIONS FOR GEOS WIND COMPUTATION
C  HERE USE PT MUGU, PT SAN BUOY, PT CONCEPTION BUOY (END OF LIST)
       IS1 = NUMNWS - 2
       IS2 = NUMNWS - 1
       IS3 = NUMNWS
C  SET NTHERM
      NTHERM = 1
      PR1 = PRESS(IS1) * (1013.3/29.92)
      PR2 = PRESS(IS2) * (1013.3/29.92)
      PR3 = PRESS(IS3) * (1013.3/29.92)
      STLT1 = STLT(IS1)
      STLT2 = STLT(IS2)
      STLT3 = STLT(IS3)
      STLN1 = STLN(IS1)
      STLN2 = STLN(IS2)
      STLN3 = STLN(IS3)
      TMP1 = TEMP(IS1)
      TMP2 = TEMP(IS2)
      TMP3 = TEMP(IS3)
C  DEFINE CONSTANTS
      AVLAT = .333* (STLT1 + STLT2 + STLT3)/57.295
      AVLON = .333* (STLN1 + STLN2 + STLN3)/57.295
      FC = 14.584 * SIN (AVLAT)
C    CORIOLIS FORCE IN UNITS 10-5 SEC-1
      COSLAT = COS (AVLAT)
      DENOM = (STLT2-STLT1) *(STLN3-STLN1) - (STLT3-STLT1) *
     +        (STLN2-STLN1)
      RHO = 1.1
```

```
C   DENSITY IN UNITS 10-3 G/CM3
         C2 = 100.0/(RHO *1.112)
C   COMPUTE GEOSTROPHIC WINDS
        DPDLT = ((STLN2 - STLN1) * (PR3-PR1) - (STLN3 -STLN1) *
     +           (PR2 - PR1))/(-DENOM)
        DPDLN = ((STLT2 - STLT1) * (PR3 - PR1) - (STLT3 -STLT1) *
     +           (PR2 - PR1))/DENOM
        UGEOS = (-C2/FC) * DPDLT
        VGEOS =  (C2/FC) * (DPDLN/COSLAT)
C   SPEED UNITS ARE M SEC-1
        PRINT 30, UGEOS, VGEOS
C   THIS PART MAKES THERMAL WIND CORRECTION TO UGEOS, VGEOS
C   NTHERM IS INDICATOR FOR USE (WHEN=1)
        IF (NTHERM .NE. 1) GO TO 200
        FVNIN  = 5.0/9.0
        TMP1 =  (TMP1 - 32.0) * FVNIN
        TMP2 =  (TMP2 - 32.0) * FVNIN
        TMP3 =  (TMP3 - 32.0) * FVNIN
        AVTMP = 273.0 + .333 *(TMP1 + TMP2 + TMP3)
C   GRAVITY = 9.8 M SEC-2, TEMP IN DEG K
        C3 = 9.8/(FC * 1.112)
        DTDLT =((STLN2 -STLN1)* (TMP3-TMP1) -(STLN3-STLN1)
     +     *(TMP2-TMP1))/(-DENOM)
        DTDLN =((STLT2-STLT1)* (TMP3-TMP1) -(STLT3-STLT1)
     +     *(TMP2-TMP1))/DENOM
        UTHERM = -(C3/AVTMP)*DTDLT
        VTHERM =  (C3/AVTMP)*(DTDLN/COSLAT)
C   ASSUME SHEAR ACTS OVER LAYER OF DEPTH DEPL (IN M)
          DEPL = 200    !  TIE THIS IN TO AVTHK FOR B LYR TOP
        USHEAR = UTHERM * DEPL
        VSHEAR = VTHERM * DEPL
         PRINT 40, USHEAR, VSHEAR, DEPL
        UGEOS = UGEOS + USHEAR
        VGEOS = VGEOS + VSHEAR
         PRINT 50
         PRINT 30, UGEOS, VGEOS
 200    CONTINUE
C**END OF GEOSTROPHIC WIND SECTION
C INTERPOLATE WINDS BETWEEN SFC AND GEOSHT
        DO 450 IT = 1, NUMNWS
        DO 400 K = 2, NLVL  ! COUNTER FOR SIGMA LEVELS
          IF (RHS1(IT,K) .GT. 0.0) GO TO 370
            UNWS(IT,K) = 0.0
            VNWS(IT,K) = 0.0
            GO TO 400
  370     IF (RHS1(IT,NLVL) .GE. 800.) GEOSHT = 800.
          IF (RHS1(IT,NLVL) .LT. 800.) GEOSHT = RHS1(IT,NLVL)
         IF (RHS1(IT,K) .GT. GEOSHT) GO TO 380
C   FOR SIGMA LEVELS BETWEEN SURFACE OBS AND GEOSHT
         RATIO=(ALOG10(RHS1(IT,K))-1.0)/(ALOG10(GEOSHT) -1.0)
         UNWS(IT,K) = UCOMP(IT) +(UGEOS-UCOMP(IT)) * RATIO
         VNWS(IT,K) = VCOMP(IT) +(VGEOS-VCOMP(IT)) * RATIO
         GO TO 400
C   FOR SIGMA LEVELS ABOVE  GEOSHT
  380    UNWS(IT,K) = UGEOS
         VNWS(IT,K) = VGEOS
  400   CONTINUE
        PRINT 9020, IT
        PRINT 9010, (UNWS(IT,K), VNWS(IT,K), K=1,NLVL)
  450   CONTINUE
C   INCLUDE NWS DATA WITH DOPPLER DATA IN ARRAYS NEEDED
```

```
C    FOR OBJECTIVE ANALYSIS.
         DO 460 N = 1, NUMNWS
            M = N + NUMDOP
           XG(M) = XG1(N)
           YG(M) = YG1(N)
         DO 460  K = 1, NLVL
            USIG(M,K) = UNWS(N,K)
            VSIG(M,K) = VNWS(N,K)
 460     CONTINUE
         NUMTOT = NUMDOP + NUMNWS
         DO 480 M = 1, NUMTOT
         PRINT 9020,M
         PRINT 9010, (USIG(M,K), VSIG(M,K), K=1,NLVL)
 480     CONTINUE
         PRINT 9003
 9002    FORMAT (//'  STA. NO.   U COMP   V COMP '/)
 9003    FORMAT  (/'     END OF SUBROUTINE GEOSIG  '/)
 9004    FORMAT (//'   LATITUDE AND LONGITUDE OF STATIONS '/)
 9005    FORMAT (/' WIND  DIR  WIND SPEED,  NWS DATA'/)
 9006    FORMAT (/' U COMPONENT V COMPONENT AT SIGMA LEVELS'/)
 9007    FORMAT (//'  THE ANCHOR PT. IS AT LAT ='F9.3,'  AND LONG='F9.3)
 9008    FORMAT (//'  X AND Y OF STATIONS IN KM AND IN COLS,ROWS
      +  FROM SW CORNER OF COARSE GRID'/)
 9010    FORMAT (3X,8F6.1)
 9011    FORMAT (/4X,2F11.0,8X,2F10.1)
 9013    FORMAT (/' STA. ='I5,' UCOMP='F8.1,'   VCOMP='F8.1/)
 9014    FORMAT (3X,I5,2F8.1)
 9015    FORMAT (3X,I5,8F8.1)
 9020    FORMAT (/' STATION NUMBER ='I6/)
 9024    FORMAT  (/'  HEIGHTS OF SIGMA SURFACES '/)
         RETURN
          END
         SUBROUTINE GPAN
C  THIS ROUTINE MAKES A GRID PT ANALYSIS FROM AVAILABLE
C  WIND OBSERVATIONS.  THE OBSERVATIONS HAVE BEEN INTERPO-
C  LATED TO SIGMA SURFACES.  THE WEIGHTING GIVEN TO EACH
C  STATION IS INVERSELY PROPORTIONAL TO ITS DISTANCE FROM
C  THE GRID POINT.
C  FOR SUBROUTINE DIABWND, JAN  '85.
C  BY R.M. ENDLICH, SRI INTN'L, MENLO PARK CA 94025.
         DIMENSION WT(50)
         COMMON /WINDS/ USIG(50,6), VSIG(50,6)
         COMMON /STALOC/ XG(50), YG(50)
         INTEGER STAID(6)
         REAL MINDIST
        COMMON/LIMITS/NCOL,NROW,NLVL,NCOLM1,NROWM1
        COMMON/UARS/U(50,26,6),UA(50,26,6),V(50,26,6),VA(50,26,6)
        COMMON/PARMS/ZTOP,DS,DSIGMA,NLVLM1,XHT1,XHT2,X1,Y1,
       1 X2,Y2,UG,VG,RATIO,TDSI
        COMMON /SITE/ IXS, JYS, THSITE, IGRID
        COMMON /ANCHOR/ SLAT, SLNG
        COMMON /NUMOBS/ NUMDOP, NUMNWS, NUMTOT
        COMMON /PRLIM/ I1, I2
C  VARIABLES ARE:
C    USIG = U COMP OF WIND ON A SIGMA SFC
C    VSIG = V COMP OF WIND ON A SIGMA SFC
C    WT = WEIGHT ASSIGNED TO A GIVEN STATION
C    U(I,J,K) AND V(I,J,K) ARE FINAL WIND COMPONENTS
C    NLVL = NUMBER OF SIGMA LEVELS
C    XG,YG ARE DISTANCES OF STATIONS FROM SW CORNER OF COARSE
C    GRID AND ARE MEASURED IN GRID UNITS (DSCRS)
```

```
C     IN ARRAYS (IT,K) IT DENOTES STATIONS, K DENOTES LEVELS
C     IN 3-D ARRAYS I,J,K DENOTE COLS, ROWS, LEVELS FROM SW CORNER
C ------------------------------
      PRINT 9001
 9001  FORMAT (/ ' BEGIN SUBROUTINE GPAN  '/)
C  SET MINIMUM DISTANCE (GRID UNITS) TO AVOID INFINITE WTS
         MINDIST = 0.15
        DO 400 I = 1, NCOL
        DO 400 J = 1, NROW
           SUMWT = 0.0
          DO 100 IT = 1, NUMTOT
           DIST = (FLOAT(I) -XG(IT))**2 + (FLOAT(J)-YG(IT))**2
           DIST = SQRT(DIST)
           IF (DIST .LE. MINDIST) DIST = MINDIST
           WT(IT) = 1.0/(DIST*DIST)
           SUMWT = SUMWT + WT(IT)
 100     CONTINUE
C  NORMALIZE WEIGHTS
          DO 120 IT = 1, NUMTOT
            WT (IT) = WT(IT)/SUMWT
 120      CONTINUE
       IF (J .NE. 31)  GO TO 125
       PRINT 9030, I, J
 9030  FORMAT (/'   WTS FOR STATIONS FOR GRID POINT X,Y='2I3)
        PRINT 9010, (WT(IT),IT=1,NUMTOT)
 125   CONTINUE
C  MAKE GRID POINT ANALYSIS USING WTS AND STA DATA
          DO 350 K = 1, NLVL
            U(I,J,K) = 0.0
            V(I,J,K) = 0.0
          DO 300 IT = 1, NUMTOT
            U(I,J,K) = U(I,J,K) + WT(IT) * USIG(IT,K)
            V(I,J,K) = V(I,J,K) + WT(IT) * VSIG(IT,K)
 300      CONTINUE
 350     CONTINUE
 400    CONTINUE
       DO 330 K = 2, NLVL
         PRINT 9035, K
 9035  FORMAT (/'  U COMPONENT AT LEVEL ='I3/)
         DO 320 JR = 1, NROW
           JP = NROW + 1 - JR
 320     PRINT 9031, (U(IP,JP,K),IP=I1,I2)
         PRINT 9038, K
 9038  FORMAT ('   V COMPONENT AT LEVEL ='I3/)
         DO 325 JR = 1, NROW
           JP = NROW + 1 - JR
 325     PRINT 9031, (V(IP,JP,K),IP=I1,I2)
 330    CONTINUE
       PRINT 9002
 9002  FORMAT (//'  END OF SUBROUTINE GPAN '/)
 9004  FORMAT (//'  LATITUDE AND LONGITUDE OF STATIONS'/)
 9006  FORMAT (/' U COMPONENT V COMPONENT AT SIGMA LEVELS'/)
 9007  FORMAT (//'  THE ANCHOR PT. IS AT LAT ='F9.3,'  AND LONG='F9.3)
 9010  FORMAT (3X,8F6.1)
 9031  FORMAT (1X,33F4.0)
        RETURN
          END
      SUBROUTINE TOPO(NUM)
C
C   READ  AND  COMPUTE TOPOGRAPHY AT GRID POINTS
C  LAST REVISION OCTOBER '84.
```

```
C  IF NUM=0 READS TERRAIN HEIGHTS FOR ALL GRIDS.
C  IF NUM .GT. 0 PICKS TERRAIN HTS , CALLS SETBLT TO ESTABLISH
C  BNDY LYR TOP, AND COMPUTES RELATIVE HTS (RHS) FOR PROPER GRID.
C
      COMMON/LIMITS/NCOL,NROW,NLVL,NCOLM1,NROWM1
      COMMON/CTOP/ MCRS,NCRS,MMED,NMED,MFIN,NFIN,NGRID
      COMMON/RARS/RHS(50,26,6)
      COMMON/CSFC/SFCHT(50,26),SIGMA(6)
      COMMON /BLHT/ BLT(50,26),HSITE, AVTHK, SLFAC,STHK,BLGRX,BLGRY
      COMMON /GROUND/ TERLIM
      COMMON /PRLIM/ I1, I2
      DIMENSION HTCRS(50,26), HTMED(50,26), HTFIN(50,26)
C  TO ACCOUNT FOR STABLE FLOWS, THE LOWEST SIGMA SFCS SHOULD
C   INTERSECT HIGH TERRAIN. THE LIMIT FOR THIS INTERSECTION IS
C   TERLIM (IN M).
    2 FORMAT(8F10.2)
    3 FORMAT(4X,-2P14F5.0)
    4 FORMAT(/,5X,21F5.0)
    6 FORMAT(4X,-2P19F5.0)
       PRINT 9001,NUM
 9001 FORMAT (/' BEGIN SUBROUTINE TOPO, NUM='I3)
      IF (NUM.GT.0) GO TO 10
C  READ TERRAIN HEIGHT VALUES AT GRID POINTS IN METERS, ALL GRIDS
       PRINT 9006
       PRINT 9003
C  IN HEIGHT DATA FILE NORTHERN ROW IS FIRST, SO INVERT ORDER.
C  READ AND PRINT HEIGHTS AT COARSE GRID POINTS
      DO 8 J=1,NCRS
       JR = NCRS + 1 -J
      READ(1,2) (HTCRS(I,JR),I=1,MCRS)
    8 CONTINUE
      DO 118 J = 1,NCRS
       JR = NCRS+1-J
C     PRINT 4, (HTCRS(I,JR),I =1,MCRS)
      PRINT 4, (HTCRS(I,JR),I =I1,I2)
  118  CONTINUE
C READ AND PRINT MEDIUM GRID HEIGHTS
       IF (NGRID .LT. 2) GO TO 120
       PRINT 9004
      DO 9 J=1,NMED
       JR = NMED + 1 -J
      READ(1,2) (HTMED(I,JR),I=1,MMED)
    9 CONTINUE
      DO 119 J = 1,NMED
      JR = NMED+1-J
      PRINT 4, (HTMED(I,JR),I =1,MMED)
  119  CONTINUE
  120  CONTINUE
C  READ AND PRINT FINE GRID HEIGHTS
       IF (NGRID .NE. 3) GO TO 214
       DO 210 J=1,NFIN
        JR = NFIN +1 -J
       READ(1,2) (HTFIN(I,JR),I=1,MFIN)
 210   CONTINUE
       PRINT 9005
       DO 212 J=1,NFIN
        JR = NFIN+1-J
       PRINT 4, (HTFIN(I,JR),I=1,MFIN)
 212  CONTINUE
 214   CONTINUE
       GO TO 150
```

```
   10 CONTINUE
      DO 15 J=1,NROW
      DO 15 I=1,NCOL
      IF (NUM .NE. 1) GO TO 11
      SFCHT(I,J)=HTCRS(I,J)
   11 CONTINUE
      IF (NUM .NE. 2) GO TO 12
      SFCHT(I,J)=HTMED(I,J)
   12 CONTINUE
      IF (NUM .NE.3) GO TO 13
      SFCHT(I,J) =HTFIN(I,J)
   13 CONTINUE
   15 CONTINUE
C** SET BNDY LAYER TOP (ARRAY BLT).
       CALL SETBLT
C DENOTE GEOMETRIC HEIGHT ABOVE TERRAIN BY RHS
      DO 67 J=1,NROW
      DO 67 I=1,NCOL
        IF (SFCHT(I,J) .LE. TERLIM) THEN
         ZVAR = BLT(I,J) - SFCHT(I,J)
         DO K=1,NLVL
         RHS(I,J,K) =SIGMA(K) * ZVAR
         END DO
        ELSE
         ZVAR = BLT(I,J) - TERLIM
         DO K=1,NLVL
         RHS(I,J,K) =SIGMA(K) * ZVAR
C  NEGATIVE VALUES OF RHS INDICATE SIGMA SFC INTERSECTS TERRAIN
         IF ((RHS(I,J,K)+TERLIM) .LT. SFCHT(I,J)) RHS(I,J,K)
     +      = -1.0
         END DO
        END IF
   67 CONTINUE
  150  PRINT 9002
 9002 FORMAT (/' END OF SUBROUTINE TOPO'/)
 9003 FORMAT (1H1,' TERRAIN HTS, COARSE GRID, METERS'/)
 9004 FORMAT (1H1,' TERRAIN HTS, MEDIUM GRID, METERS'/)
 9005 FORMAT (1H1,'  TERRAIN HTS, FINE GRID, METERS')
 9006 FORMAT (/'   PRINTOUT IS REVERSE OF INPUT - HAS NORTH ROW 1ST'/)
       RETURN
      END
      SUBROUTINE SETBLT
C** THIS SUBROUTINE SETS THE HEIGHT OF THE BNDY LAYER TOP.  AVTHK IS
C  AVER. BL THICKNESS OVER AREA. SLFAC CONTROLS THE SLOPE, IF 0 THE
C  TOP IS FLAT, IF 1 THE BL TOP FOLLOWS THE TERRAIN.  HSITE IS HT OF
C  THE ANCHOR POINT (SITE), STHK IS THE SMALLEST BL THICK ALLOWED.
C  BLGRX IS B LYR HT GRADIENT TO E.
C  BY R. M. ENDLICH SRI INTNL
C  LAST REVISION JUNE '84
      DIMENSION B(50,26),IB(50,26)
      COMMON/LIMITS/NCOL,NROW,NLVL,NCOLMI,NROWM1
        COMMON/CVOS/ RCM,RMF,IV,DSCRS,IXCRS,JYCRS,IXMED,JYMED
      COMMON /SITE/ IXS, JYS, THSITE, IGRID
      COMMON /BLHT/ BLT(50,26),HSITE, AVTHK, SLFAC,STHK,BLGRX,BLGRY
      COMMON/CSFC/ SFCHT(50,26), SIGMA(6)
      COMMON /PRLIM/ I1, I2
      PRINT 9001
      THK = AVTHK
      IF (IGRID .GT. 1) THK = THSITE
        IF (IGRID .NE.1) GO TO 5
         BLX =BLGRX
```

```
        BLY =BLGRY
        IX = IXCRS
        JY = JYCRS
  5     CONTINUE
        IF (IGRID .NE. 2) GO TO 6
        BLX= BLGRX/(RCM)
        BLY= BLGRY/(RCM)
        IX = IXMED
        JY = JYMED
  6     CONTINUE
        IF (IGRID .NE.3) GO TO 7
        BLX= BLGRX/(RCM*RMF)
        BLY= BLGRY/(RCM*RMF)
        IX = IXFIN
        JY = JYFIN
  7     CONTINUE
        ITER = 0
 10     ITER = ITER + 1
        SUM1 = 0.0
        Q1 = 0.0
        DO 50  I =1,NCOL
        DO 50 J = 1,NROW
          BLT(I,J) = THK + (SLFAC * SFCHT(I,J)) + (1.0 -SLFAC) *HSITE
C ADD EASTWARD GRADIENT TO BLT FROM COL.  OF ANCHOR POINT
          BLT(I,J)= BLT(I,J)+ (I-IX)*(BLX/FLOATJ(NCOL))
C ADD NORTHWARD GRADIENT FROM ROW  OF ANCHOR POINT
          BLT(I,J)= BLT(I,J) + (J-JY)*(BLY / FLOATJ(NROW))
          IF (SFCHT(I,J) .GT. (BLT(I,J)-  STHK )) BLT(I,J) = SFCHT(I,J)
     +         + STHK
        SUM1 = SUM1 + (BLT(I,J) - SFCHT(I,J))
        Q1 = Q1 + 1.0
 50     CONTINUE
        IF(IGRID.GT.1) GO TO 51
        ATH = SUM1/Q1
        THK = THK + (AVTHK - ATH)
        THSITE = BLT(IXS,JYS) - SFCHT(IXS,JYS)
        PRINT 9010, AVTHK,ATH, THSITE
        DIFF = ABS(AVTHK - ATH)
        IF (ITER .GT.  9) GO TO 52
        IF (IGRID .EQ. 1 .AND. DIFF .GT. 1.0) GO TO 10
 51     CONTINUE
        IF(IGRID.EQ.1) GO TO 52
C  MAKE BL THICKNESS AT ANCHOR POINT (THSITE) THE SAME FOR
C  MED AND FINE GRIDS AS IT WAS FOR COARSE GRID
        ATH=SUM1/Q1
        THSITE2=BLT(IXS,JYS) - SFCHT(IXS,JYS)
        THK=THK + (THSITE - THSITE2)
        DIFF2=ABS(THSITE - THSITE2)
        PRINT 9010, AVTHK,ATH,THSITE2
        IF(ITER.GT.9) GO TO 52
        IF(DIFF2.GT.1.0) GO TO 10
 52     CONTINUE
        DO 55  JP = 1,NROW
        DO 55  IP = 1,NCOL
          B(IP,JP) =    BLT(IP,JP)
          IB(IP,JP)= JNINT(B(IP,JP))
 55     CONTINUE
        PRINT 9115,IGRID
        DO 60  JP = 1,NROW
        JR = NROW+ 1 -JP
 60     PRINT 9105,  ( IB(IP,JR),IP=I1,I2)
```

```
      DO 65 JP= 1,NROW
      DO 65 IP= 1,NCOL
        B(IP,JP)=  (BLT(IP,JP) -SFCHT(IP,JP))
        IB(IP,JP)= JNINT(B(IP,JP))
   65   CONTINUE
      PRINT 9020
      DO 70 JP= 1,NROW
       JR= NROW +1 -JP
  70    PRINT 9105, (IB(IP,JR),IP=I1,I2)
      PRINT 9002
 9001 FORMAT (1H1,'     BEGIN SUBROUTINE SETBLT'/)
 9002 FORMAT (//'     END OF SUBROUTINE SETBLT'/)
 9010 FORMAT  ('  INITIAL AV. THICKNESS, M='F10.1,
     +  '    ACTUAL AV. THICKNESS ='F10.1,
     + '    SITE THICKNESS ='F10.1/)
 9020 FORMAT(1H1,'BNDY LAYER THICKNESS IN M'/)
 9100 FORMAT (/5X,22F5.0)
 9105 FORMAT (/1X,33I4)
 9115 FORMAT (1H1,'    HEIGHT OF BNDY LAYER TOP, M, GRID ='I3 )
        RETURN
        END
      SUBROUTINE BAL5(NITER)
C*****THIS IS VERSION 11, OCTOBER '84.
C  THIS ROUTINE BALANCES DIVERGENCE TO VALUES IN ARRAY DD (OR
C  TO DDIJ=0) AND VORTICITY TO ARRAY VT(I,J).
C  FOR WIND SPEED IN MPS. DIV AND VORT ARE SCALED TO UNITS
C  10 -6 SEC-1. THE METHOD USES DIRECT VECTOR ALTERATIONS.
C  THIS FORM FOR RECTANG GRID OMITS TRIG FUNCTIONS.
C  THE FLUX FORMULATION IS USED FOR FINITE DIFFERENCES. FOR SIGMA LAYERS
C  COMPUTE NON-DIV WINDS FOR WINDS WEIGHTED BY THICKNESS OF LAYER.
C  ASSUME SIGMADOT =0. INDICES IN ARRAYS (I,J,K) ARE I=COLUMN,
C  J=ROW, K=LEVEL; PT (1,1,1) IS AT SW CORNER AT GROUND.
C  FOR COMPUTATION BOXES, INDICES REFER TO SW CORNER OF BOX.
C  IVORT CONTROLS USE OF VORTICITY. IF=2 VORT IS NOT HELD
C  CONSTANT.
C  BY R.M. ENDLICH, SRI INTN'L, 1ST VERSION FOR LAYERS JUNE '82.
      DIMENSION VT(50,26),DI(50,26),V0(50,26),U1(50,26),V1(50,26)
      DIMENSION UN(50,26),VN(50,26),THK(50,26)
      DIMENSION IFXPT(50,26)
      COMMON /CSFC/ SFCHT(50,26),SIGMA(6)
      COMMON /UARS/U(50,26,6),UA(50,26,6),V(50,26,6),VA(50,26,6)
      COMMON/PARMS/ZTOP,DS,DSIGMA,NLVLM1,XHT1,XHT2,X1,Y1,
     +   X2,Y2,UG,VG,RATIO,TDSI
      COMMON /CVOS/ RCM,RMF,IV,DSCRS,IXCRS,JYCRS,IXMED,
     +   JYMED,ISFIX,JYFIN
      COMMON /CTOP/ MCRS, NCRS, MMED, NMED, MFIN, NFIN, NGRID
      COMMON /LIMITS/NCOL,NROW,NLVL,NCOLM1,NROWM1
      COMMON /SITE/ IXS, JYS, THSITE, IGRID
      COMMON /RARS/RHS(50,26,6)
      COMMON /PRLIM/ I1, I2
       IVORT = 2     !  IGNORE VORTICITY
      PRINT 9002
      GS = DS * 1.0E-05
C  USE GRID SPACING IN 100'S OF KM.  DS IS IN M.
      GSI = 10.0/GS         ! FOR PROPER SCALING
C  FOR RECTANGULAR GRID OMIT TRIG FUNCTIONS USED PREVIOUSLY
C  ASSIGN PTS WHERE INITIAL WIND ANALYSIS IS HELD FIXED
      IF (IGRID .NE. 1) GO TO 10   ! FOR COARSE GRID
      CALL SETDBLARRY(0, IFXPT, NCOL, NROW)
C       IFXPT(11,8) =1
 10     CONTINUE
```

```
        IF (IGRID .NE. 2) GO TO 15   ! FOR MEDIUM GRID
        CALL SETDBLARRY(0, IFXPT, NCOL, NROW)
C       IFXPT(  ,  ) = 1
  15    CONTINUE
C  PRINT POINTS WITH FIXED WINDS
        DO 25 J = 1,NROW
        DO 25 I = 1,NCOL
          IF (IFXPT(I,J) .EQ. 1) PRINT 9582, I,J
  25    CONTINUE
C  ARRAYS U,V,RHS ARE WRITTEN (COLS,ROWS,LEVELS)
C  TRANSFER WINDS TO 2D ARRAYS
        DO 800 L=2,NLVL
                DO 35 J=1,NROW
                DO 35 I=1,NCOL
                        UN(I,J)= U(I,J,L)
                        VN(I,J)= V(I,J,L)
   35           CONTINUE
        CALL SETDBLARRY(0.0,DI ,NCOL,NROW)
        CALL SETDBLARRY(0.0,VT ,NCOL,NROW)
        CALL SETDBLARRY(0.0,VO ,NCOL,NROW)
C.COMPUTE LAYER THICKNESS AND MULTIPLY WIND COMPONENTS
       DO 40 J =1,NROW
       DO 40 I =1,NCOL
         LA = L +1
         IF (LA .GT. NLVL) LA = NLVL
         HTA = RHS(I,J,LA)
         LB = L -1
         IF (LB .LT. 1) LB =1
         HTB = RHS(I,J,LB)
         THK(I,J) = 0.5 *(HTA -HTB) * 0.01
        IF (THK(I,J) .LE. 0.) THK(I,J) =1.0  ! FOR NEG. RHS
C UNITS OF THICKNESS ARE HUNDREDS OF M FOR CONVENIENCE
C  SET INITIAL WINDS BEFORE ALTERATIONS
         U1(I,J) = UN(I,J)
         V1(I,J) = VN(I,J)
C  WEIGHT WINDS WITH THICKNESS OF LAYER
        UN(I,J) = U1(I,J) * THK(I,J)
        VN(I,J) = V1(I,J) * THK(I,J)
 40    CONTINUE
C       PRINT 9520
C       DO 45 J =1,NROW
         JP = NROW+1-J
C 45    PRINT 9102, (THK(I,JP),I =I1,I2)
C       COMPUTE ORIGINAL DIVERGENCE AND VORTICITY
        DO 170 J =1,NROWM1
        DO 170 I =1,NCOLM1
         UE = 0.5 * (UN(I+1,J) + UN(I+1,J+1))
         UW = 0.5 * (UN(I,J)   + UN(I,J+1))
         VSO = 0.5 * (VN(I+1,J) + VN(I,J))
         VNO= 0.5 * (VN(I,J+1) + VN(I+1,J+1))
         DUE = GSI  * (UE - UW)
         DVN = GSI * (VNO - VSO)
         DI(I,J) = DUE + DVN    !  DIV, UNITS ARE 10-6 SEC-1
         IF (IVORT .EQ. 2) GO TO 170
         VE = 0.5 * (VN(I+1,J) + VN(I+1,J+1))
         VW = 0.5 * (VN(I,J)   + VN(I,J+1))
         USO = 0.5 * (UN(I+1,J) + UN(I,J))
         UNO = 0.5 * (UN(I,J+1) + UN(I+1,J+1))
         DVE = GSI  * (VE - VW)
         DUN = GSI * (UNO - USO)
         VT(I,J) = DVE - DUN      !  VORTICITY
```

```
 170   CONTINUE
C        PRINT 9570
C       DO 173 J =1,NROW
         JP = NROW+1-J
C 173   PRINT 9100, (VT(I,JP),I =I1,I2)
C       PRINT 9571
C       DO 172 J =1,NROW
         JP = NROW+1-J
C 172   PRINT 9100, (DI(I,JP),I =I1,I2)
C  FOR NONDIVERGENCE SET DDIJ =0.0
         DDIJ = 0.0
          LG = 0
      RA = 0.4    !  RELAXATION FACTOR
 210  LG = LG+1
      DO 500 J =1,NROWM1
      DO 500 I =1,NCOLM1
        UE = 0.5 * (UN(I+1,J) + UN(I+1,J+1))
        UW = 0.5 * (UN(I,J)   + UN(I,J+1))
        VSO = 0.5 * (VN(I+1,J) + VN(I,J))
        VNO= 0.5 * (VN(I,J+1) + VN(I+1,J+1))
        DUE = GSI  * (UE - UW)
        DVN = GSI * (VNO - VSO)
        DI(I,J) = DUE + DVN
        CUIJ = .05 * GS  *(DDIJ -DI(I,J)) *RA
        CVIJ = .05 *GS * (DDIJ - DI(I,J)) * RA
        IF (CUIJ .LT. -1.0) CUIJ = -1.0  !  LIMIT CHANGES
        IF (CUIJ .GT.  1.0) CUIJ =  1.0
        IF (CVIJ .LT. -1.0) CVIJ = -1.0
        IF (CVIJ .GT.  1.0) CVIJ =  1.0
        UN(I+1,J) = UN(I+1,J) + CUIJ
        UN(I+1,J+1) = UN(I+1,J+1) +CUIJ
        UN(I,J)  = UN(I,J)  -CUIJ
        UN(I,J+1) = UN(I,J+1)  -CUIJ
        VN(I+1,J) = VN(I+1,J) - CVIJ
        VN(I,J)  = VN(I,J) - CVIJ
        VN(I,J+1) = VN(I,J+1) + CVIJ
        VN(I+1,J+1) = VN(I+1,J+1) + CVIJ
        IF (IVORT .EQ. 2) GO TO 490
        VE = 0.5 * (VN(I+1,J) + VN(I+1,J+1))
        VW = 0.5 * (VN(I,J) + VN(I,J+1))
       USO = 0.5 * (UN(I+1,J) + UN(I,J))
      UNO= 0.5 * (UN(I,J+1) + UN(I+1,J+1))
      DVE = GSI  * (VE - VW)
      DUN = GSI * (UNO - USO)
      VO(I,J) = DVE - DUN
      CVIJ = .05 * GS  *(VT(I,J) -VO(I,J)) *RA
      CUIJ = .05 *GS *(VT(I,J) - VO(I,J)) * RA
       IF (CUIJ .LT. -1.0)  CUIJ = -1.0  ! LIMIT CHANGES
       IF (CUIJ .GT.  1.0) CUIJ =  1.0
       IF (CVIJ .LT. -1.0) CVIJ = -1.0
       IF (CVIJ .GT.  1.0) CVIJ =  1.0
      UN(I,J)  = UN(I,J)  + CUIJ
      UN(I+1,J) = UN(I+1,J) + CUIJ
      UN(I,J+1) = UN(I,J+1)  - CUIJ
      UN(I+1,J+1) = UN(I+1,J+1) - CUIJ
      VN(I+1,J) = VN(I+1,J) + CVIJ
      VN(I+1,J+1) = VN(I+1,J+1) +CVIJ
      VN(I,J)  = VN(I,J) - CVIJ
      VN(I,J+1) = VN(I,J+1) - CVIJ
 490    CONTINUE
C  TO KEEP WINDS 0.0 WHERE RHS IS NEGATIVE
```

```
        IF (RHS(I,J,L) .LT. 0.0) THEN
         UN(I,J) = 0.0
         VN(I,J) = 0.0
        END IF
C  HOLD ANALYZED WIND COMPONENTS FIXED AT PTS WHERE IFXPT =1
        IF (IFXPT(I,J)  .NE. 1) GO TO 500
         UN(I,J) = U1(I,J) * THK(I,J)
         VN(I,J) = V1(I,J) * THK(I,J)
 500     CONTINUE
         IF (LG .GT. NITER) GO TO 540
       DO 530 J=2,NROWM1
       DO 530 I =2,NCOLM1
          IF (ABS(DDIJ -DI(I,J)) .GT. 50.0) GO TO 210
C        IF (ABS(VT(I,J) -VO(I,J)) .GT. 50.0) GO TO 210
 530   CONTINUE
 540   CONTINUE
C                 PRINT 9570
C         PRINT 9580
C        DO 510 J=1,NROW
         JP = NROW+1-J
C 510    PRINT 9100,    (VO(I,JP),I =I1,I2)
C        PRINT 9200, LG
C         PRINT 9571
C        PRINT 9580
C        DO 520 J=1,NROW
         JP = NROW+1-J
C 520    PRINT 9100,    (DI(I,JP),I =I1,I2)
         SUM1=0.0
         SUM2=0.0
         Q1=0.0
        DO 1040 J =1,NROW
        DO 1040 I =1,NCOL
        IF (RHS(I,J,L) .LE. 0.0) GO TO 1040   !  OMIT THESE PTS
          Q1 = Q1 + 1.0
          UN(I,J) = UN(I,J)/THK(I,J)
          VN(I,J) = VN(I,J)/THK(I,J)
          U1(I,J) = U1(I,J) - UN(I,J)
          V1(I,J) = V1(I,J) - VN(I,J)
          SUM1=SUM1 + U1(I,J)
          SUM2=SUM2 + V1(I,J)
 1040 CONTINUE
         SUM1 = SUM1/Q1
         SUM2 = SUM2/Q1
C NORMALIZE ORIG. AVERAGE VALUES
C       DO 1045 J =1,NROW
C       DO 1045 I =1,NCOL
C          UN(I,J) = UN(I,J) + SUM1
C          VN(I,J) =VN(I,J) +SUM2
 1045    CONTINUE
         PRINT 1145, L
 1145    FORMAT(' U COMP. DIVERGENT, LEVEL ='I3 )
         DO 1160 J=1,NROW
         JP = NROW+1-J
 1160    PRINT 9150, (U1(I,JP),I =I1,I2)
 9150    FORMAT(' ',33F4.1)
         PRINT 1155
 1155    FORMAT(' V COMP. DIVERGENT ')
         DO 1170 J=1,NROW
         JP = NROW+1-J
 1170    PRINT 9150, (V1(I,JP),I =I1,I2)
C  WRITE LEVEL 4 DIV WINDS TO OUTPUT FILE
```

```
      IF (L .NE. 4) GO TO 700
         DO 690 J = 1, NROW
           JR = NROW + 1 -J
 690     WRITE (3,9065) (U1(I,JR),I=1,NCOL)
         DO 695 J = 1, NROW
           JR = NROW + 1 -J
 695     WRITE (3,9065) (V1(I,JR),I=1,NCOL)
 700   CONTINUE
C
C  CHANGE BACK TO 3D ARRAYS
C
       DO 580 J=1,NROW
       DO 580 I =1,NCOL
              U(I,J,L)= UN(I,J)
              V(I,J,L)= VN(I,J)
  580  CONTINUE
  800  CONTINUE
      PRINT 9003
 8888 CONTINUE
 9002 FORMAT (//'  BEGIN SUBROUTINE BAL5 '/)
 9003 FORMAT (//'  END OF SUBROUTINE BAL5 '/)
 9065  FORMAT (10F8.1)
 9100 FORMAT  (/1X,33F4.0)
 9102 FORMAT (/1X,33F4.1)
 9200 FORMAT (/'  NUMBER OF ITERATIONS ='I5)
 9520 FORMAT (1H1,'  LAYER THICKNESS IN HUNDREDS OF M'/)
 9560 FORMAT (15H DIVERGENT WIND/)
 9561 FORMAT (19H NON DIVERGENT WIND/)
 9565 FORMAT (1H1,12H U COMPONENT)
 9566 FORMAT (1H1,12H V COMPONENT)
 9581 FORMAT (16H ANALYZED VALUES/)
 9570 FORMAT (1H1,30H RELATIVE VORTICITY 10-6 SEC-1)
 9571 FORMAT (1H1,22H DIVERGENCE 10-6 SEC-1)
 9580 FORMAT (16H BALANCED VALUES/)
 9582 FORMAT (/' POINT WITH FIXED WIND IS COL ='I3,'ROW ='I3/)
      RETURN
      END
       SUBROUTINE SETDBLARRY(VALUE,ARRAY,NUM1,NUM2)
C
C      INITIALIZES ALL ELEMENTS OF ARRAY TO VALUE
C      REVISON SEPT. 1978
C
       DIMENSION ARRAY(NUM1,NUM2)
       DO 10 I=1,NUM1
       DO 10 J=1,NUM2
              ARRAY(I,J)=VALUE
   10  CONTINUE
       RETURN
       END
```

Appendix B

PROGRAM LISTING FOR MADICT

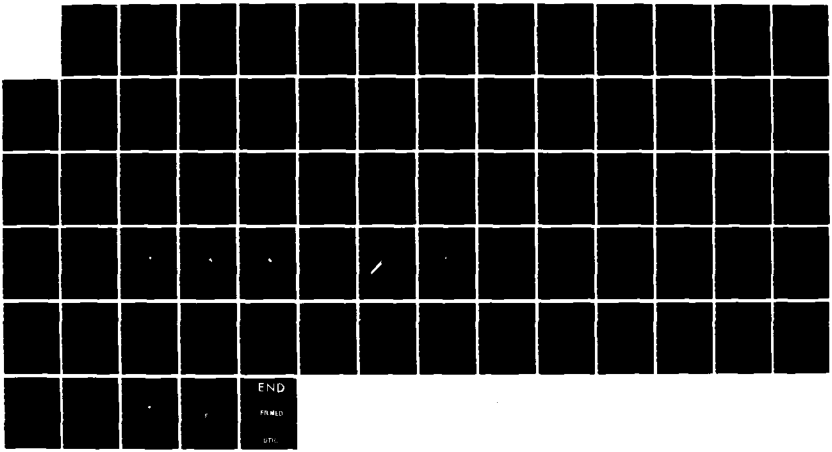
Subroutines:

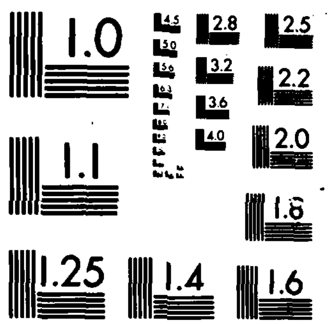
- ADVECT
- ADV 1ST
- ADV 2ND
- BASICS
- CALCON
- DEFINE
- DOTINT
- EVCO
- FILEID
- GETWND
- GRAFF
- GRIDXY
- INFLU
- INWIND
- MET IN
- MRG PRG
- NEXTHR
- OUTNOW
- RELHT
- VIRTVL
- WIND
- WINDDO

```
        PROGRAM MADICT
C
C       (Model of Atmospheric Diffusion In Complex Terrain)
C
C  BY F. L. LUDWIG, SRI INT'L, MENLO PK., CA 94025--PHONE:(415)859-2915
C THIS INTERACTIVE PROGRAM IS A PUFF SIMULATION MODEL FOR A SINGLE PT.
C SOURCE UNDER CHANGING METEOROLOGICAL CONDITIONS. WIND FIELD IS GENERATED
C FROM PRECALCULATED, NONDIVERGENT SOL'NS TO A SET OF NORMALIZED LINEARLY
C INDEPENDENT INPUTS & A MEAN SET OF INPUTS. OCTOBER 1985 VERSION.
C
        LOGICAL LOUTFL,KEY0,QUIT,DBUG
        INTEGER SL
      PARAMETER   (KEY0=.FALSE.)
      PARAMETER   (INDX=9,INDY=9,INDZ=5,NEIGN=9,NSTA=8)
      PARAMETER   (NRCPTX=20,NRCPTY=20,MXNPF=199)
      COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
     $                DOTX(0:1,0:NEIGN)
      COMMON /WNDEIG/ UM(0:NSTA+1),VM(0:NSTA+1),
     $                   UE(0:NEIGN,0:NSTA+1),VE(0:NEIGN,0:NSTA+1)
      COMMON /PFFCON/ PUFINT(0:MXNPF,0:4),PFFPAR(0:MXNPF,0:2),RSPACM,
     $                CONCEN(0:NRCPTX,0:NRCPTY,0:5),GSIGZ(6),
     $                ALPHAZ(6),ALPHAY(6),FEXP(0:90),RCPTRX(0:NRCPTX),
     $                RCPTRY(0:NRCPTY),XRCPO,YRCPO
      COMMON /WNDFLD/ WNDUVW(0:INDX,0:INDY,0:INDZ,0:2),GSPACE,
     $                PRECAL(0:INDX,0:INDY,0:1,0:2,0:NEIGN),
     $                RHS(0:INDX,0:INDY,2),XORIGN,YORIGN,ZORIGN
      COMMON /SOURCE/ XSOURC,YSOURC,ZSOURC,QSOURC
      COMMON /LOGES/  LOUTFL,DBUG
      DIMENSION VDISR0(6),VDISZ0(6)
      DATA NHOUR,SL,DOTP(0)  /2*-1,1.0/
      DATA SIGY0,SIGZ0,GSPACE,NOPUFF,GSIGY
     $     /40.0,  3.,    6000.0,  0,    0.9/
C
C SET DBUG=.TRUE. TO GET PRINTOUT HELPFUL IN CHECKING PROGRAM
C
      DATA  XORIGN, YORIGN, ZORIGN, DBUG
     $     / -30000.,-30000., 0.0, .FALSE./
      DATA ALPHAY /0.428, 0.301, 0.198, 0.128, 0.095, 0.065 /
      DATA ALPHAZ /0.025, 0.075, 0.112, 0.196, 0.231, 0.227 /
      DATA GSIGZ  /1.38,  1.07,  0.91,  0.72,  0.64,  0.575/
C
C       INPUT BASIC INFORMATION & MAKE SOME REQUIRED CALCULATIONS
C
        CALL BASICS  (VDISR0,VDISZ0,IHOUR)
C
C       GETTING READY FOR NEXT HOUR'S MET'L'G'CAL INPUTS--ONLY SET UP
C       TO RUN FOR 6 HOURS  (0 TO 5).
C
410   PRINT *, 'FOR HR FROM ',IHOUR,' TO ',IHOUR+1
      NHOUR=NHOUR+1
       IF  (NHOUR.GT.5) GO TO 86
C
C INPUT METEOROLOGICAL INFORMATION
C
        CALL MET IN  (QUIT,ISTAB,IHOUR,ALPHAY,SEPMAX)
C
C       IF FLAG SET IN MET IN, QUIT
C
        IF (QUIT) GO TO 86
C
C        IDENTIFY APPROPRIATE FILE AND READ WIND FIELD SOL'NS
C
      CALL WINDDO
C
C ---------------- START TIME LOOP-----------------------------------------
C
```

AD-A163 819 A MODEL OF AIRFLOW AND DIFFUSION IN COMPLEX TERRAIN (MADICT)(U) SRI INTERNATIONAL MENLO PARK CA F L LUDWIG NOV 85 ARO-19630.5-GS DAAG29-83-K-0009 2/2

UNCLASSIFIED F/G 4/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
      DO 1270 ITIME=0, 5
C
C  INTERPOLATING INNER PRODUCTS TO THIS TIME STEP
C
              CALL DOTINT (ITIME)
C
C        IDENTIFY CELLS WHERE WIND IS UNCALCULATED YET
C
              CALL WNDBAR(WNDUVW,YORIGN)
C
C        IF STABILITY CHANGED, RECALCULATE VIRTUAL TRAVEL
C
              IF (SL.NE.ISTAB)  CALL VIRTVL(ISTAB,SL)
C
C        FLAG FOR OUT-OF-BOUNDS PUFFS
C
              LOUTFL=.FALSE.
C
C        ADVECT NEWLY GENERATED PUFFS
C
              CALL ADV 1ST (LASTPF,VDISR0,VDISZ0,ISTAB)
C
C       ADVECT ALREADY EXISTING PUFFS
C
              CALL ADV 2ND
C
C        MERGE AND PURGE PUFFS
C
              CALL MRG PRG (LASTPF,SEPMAX)
C
C ---------------------- END OF TIME LOOP-----------------------
C
1270   CONTINUE
C
C       ZEROING CONCENTRATIONS AT RECEPTORS
        DO 1330 I=0,NRCPTX
        DO 1330 J=0,NRCPTY
          CONCEN(I,J,NHOUR)=0
1330   CONTINUE
C
C        DETERMINE AFFECTED RECEPTORS FOR EACH PUFF & ACCUMULATE
C        CONCENTRATIONS
C
        CALL INFLU (GSIGY,NHOUR,ISTAB)
C
C  GRAPH OUTPUT
C
        CALL GRAFF (NHOUR,DBUG)
C
C  SAVING PRECEDING STABILITY FOR COMPARISON WITH NEXT HOUR'S
C
      SL=ISTAB
      GO TO 410
C
C  PRINTING RESULTS
C
 86   CALL OUTNOW(NHOUR,SL)
        END
```

```
      SUBROUTINE ADVECT(XX,YY,ZZ,TRAVEL)
C
C  THIS SUBROUTINE ADVECTS PUFFS IN 5 2-MIN TIME STEPS
C
        LOGICAL LOUTFL,DBUG

        PARAMETER (NEIGN=9,INDX=9,INDY=9,INDZ=5)
      COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
     $                DOTX(0:1,0:NEIGN)
      COMMON /WNDFLD/ WNDUVW(0:INDX,0:INDY,0:INDZ,0:2),GSPACE,
     $                PRECAL(0:INDX,0:INDY,0:1,0:2,0:NEIGN),
     $                RHS(0:INDX,0:INDY,2),XORIGN,YORIGN,ZORIGN
      COMMON /LOGES/ LOUTFL,DBUG
       TRAVEL=0
       DO 2590 ISTEP=0,4
          CALL WIND(XX,YY,ZZ,U1,U2,U3)
C
C  CHECK FOR OUT OF BOUNDS
C
          IF (LOUTFL) GO TO  2600
          XX=XX+U1*0.2
          YY=YY+U2*0.2
          CALL INDXY(XX,YY,LX,LY,LOUTFL)
          ZZ=ZZ+U3*0.2
          IF (LOUTFL) GO TO 2590
          IF (ZZ.LT.RHS(LX,LY,1))  ZZ=RHS(LX,LY,1)
          IF (ZZ.GT.RHS(LX,LY,2))  ZZ=RHS(LX,LY,2)
          TRAVEL=TRAVEL+0.2*SQRT (U1*U1+U2*U2)
2590   CONTINUE
2600   RETURN
       END
C
C*************************************************************************
C
        SUBROUTINE ADV 1ST (LASTPF,VDISR0,VDISZ0,ISTAB)
C
C       THIS SUBROUTINE ADVECTS THE PUFFS WHEN THEY ARE 1ST GENERATED
C       --TAKING INTO ACCOUNT THAT MOST DO NOT MOVE FOR THE ENTIRE
C       TIME STEP.
C
        LOGICAL DBUG,LOUTFL
        PARAMETER (MXNPF=199,NEIGN=9)
        PARAMETER (NRCPTX=20,NRCPTY=20,INDX=9,INDY=9,INDZ=5)
      COMMON /WNDFLD/ WNDUVW(0:INDX,0:INDY,0:INDZ,0:2),GSPACE,
     $                PRECAL(0:INDX,0:INDY,0:1,0:2,0:NEIGN),
     $                RHS(0:INDX,0:INDY,2),XORIGN,YORIGN,ZORIGN
      COMMON /SOURCE/ XSOURC,YSOURC,ZSOURC,QSOURC
      COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
     $                DOTX(0:1,0:NEIGN)
      COMMON /PFFCON/ PUFINT(0:MXNPF,0:4),PFFPAR(0:MXNPF,0:2),RSPACM,
     $                CONCEN(0:NRCPTX,0:NRCPTY,0:5),GSIGZ(6),
     $                ALPHAZ(6),ALPHAY(6),FEXP(0:90),RCPTRX(0:NRCPTX),
     $                RCPTRY(0:NRCPTY),XRCPO,YRCPO
      COMMON /LOGES/  LOUTFL,DBUG
      DIMENSION VDISR0(6),VDISZ0(6)
      IF (NOPUFF.LE.1) THEN
         CALL INDXY(XSOURC,YSOURC,LX,LY,LOUTFL)
         IF (LOUTFL) STOP 'SOURCE OUT OF BOUNDS'
         ZSOURC=ZSOURC+RHS(LX,LY,1)
      END IF
      CALL WIND(XSOURC,YSOURC,ZSOURC,U1,U2,U3)
C
C  WNSPD UNITS--M/(10-MIN).SEPNI SET SO THAT SEPARATION IS APPROX
C  = TO SIGMAY AFTER 3 MINUTES, OR 20 METERS, WHICHEVER IS SMALLER.
C
      WNDSPD=SQRT(U1*U1+U2*U2)
      SEPNI=(ALPHAY(ISTAB)*(VDISR0(ISTAB)+0.3*WNDSPD)**0.9)
      IF (SEPNI.LT.20.0) SEPNI=20.0
```

```
C  GET WND AT SRCE & NO NEW PFS (SEPNI SPCNG)FOR 10-MIN PERIOD.
C  SEE SUBR. DEFINE FOR PUFF PARAM. DEFINITIONS
C
      NEWPFF=1.0+(WNDSPD/SEPNI)
      IF(DBUG) PRINT*,'SPD AT SRCE & OUTFLG',SQRT(U1*U1+U2*U2),LOUTFL
      NPUFFS=MXNPF-NOPUFF
      IF(DBUG) PRINT*,'********** NO NEW&OLD ', NEWPFF,NOPUFF
      IF (NEWPFF.LT.NPUFFS)  NPUFFS=NEWPFF
      NEXT=NOPUFF+1
      LASTPF=NOPUFF+NPUFFS
C
C  GETTNG INITIAL VALUES
C
      DO 900 IPUFF=NEXT,LASTPF
         PFFPAR(IPUFF,0)=600.0*QSOURC/NPUFFS
         PFFPAR(IPUFF,1)=VDISR0(ISTAB)
         PFFPAR(IPUFF,2)=VDISZ0(ISTAB)
         PUFINT(IPUFF,0)=DEPTH
         PUFINT(IPUFF,4)=0
 900  CONTINUE
      X0=XSOURC
      Y0=YSOURC
      Z0=ZSOURC
      TRAVEL=0.0
      CALL DEFINE(LASTPF,X0,Y0,Z0,TRAVEL)
      PUFINT(LASTPF,4)=0.0
      TSTEP=1.0/NPUFFS
      DO 950 IPUFF=LASTPF-1,NEXT,-1
         CALL INDXY(X0,Y0,LX,LY,LOUTFL)
         IF (.NOT.LOUTFL) THEN
                CALL WIND(X0,Y0,Z0,U1,U2,U3)
                WNDSPD=SQRT(U1*U1+U2*U2)
                X0=X0+U1*TSTEP
                Y0=Y0+U2*TSTEP
                Z0=Z0+U3*TSTEP
                TRAVEL=TRAVEL+WNDSPD*TSTEP
C
C  DEFINE PUFF PARAMETERS AFTER ADVECTION.
C
                CALL DEFINE(IPUFF,X0,Y0,Z0,TRAVEL)
                AGE=AGE+TSTEP
                PUFINT(IPUFF,4)=AGE
         ELSE
                PFFPAR(IPUFF,0)=0.0
         END IF
         IF(Z0.GT.RHS(LX,LY,2)) Z0=RHS(LX,LY,2)
         IF(Z0.LT.RHS(LX,LY,1)) Z0=RHS(LX,LY,1)
           IF(DBUG.AND.MOD(IPUFF,15).EQ.1) PRINT6000,IPUFF,LOUTFL,
     $     (PUFINT(IPUFF,JJP),JJP=0,4),(PFFPAR(IPUFF,KKP),KKP=0,2)
6000    FORMAT(' PUFF-STUFF (ADV1ST)--#,OUT?,MAXMIX,X,Y,Z,AGE,Q,VTZ,VTX'
     $        ,1X,I2,L3,6F8.0,2E12.2)
950   CONTINUE
        RETURN
        END
```

```
C
      SUBROUTINE ADV 2ND
C
C     THIS SUBROUTINE ADVECTS OLD PUFFS FOR A TIME STEP.
C
      LOGICAL DBUG
      PARAMETER (NEIGN=9,MXNPF=199,NRCPTX=20,NRCPTY=20)
      PARAMETER (INDX=9,INDY=9,INDZ=5)
     COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
    $                DOTX(0:1,0:NEIGN)
     COMMON /PFFCON/ PUFINT(0:MXNPF,0:4),PFFPAR(0:MXNPF,0:2),RSPACM,
    $                CONCEN(0:NRCPTX,0:NRCPTY,0:5),GSIGZ(6),
    $                ALPHAZ(6),ALPHAY(6),FEXP(0:90),RCPTRX(0:NRCPTX),


    $                RCPTRY(0:NRCPTY),XRCPO,YRCPO
     COMMON /WNDFLD/ WNDUVW(0:INDX,0:INDY,0:INDZ,0:2),GSPACE,
    $                PRECAL(0:INDX,0:INDY,0:1,0:2,0:NEIGN),
    $                RHS(0:INDX,0:INDY,2),XORIGN,YORIGN,ZORIGN
     COMMON /LOGES/  LOUTFL,DBUG
     DO 1000 IPUFF=0,NOPUFF
C
C  ADVECTNG OLD PFS
C
      IF (PFFPAR(IPUFF, 0).EQ. 0) GO TO 1000
      XX=PUFINT(IPUFF, 1)
      YY=PUFINT(IPUFF, 2)
      ZZ=PUFINT(IPUFF, 3)
      CALL INDXY(XX,YY,LX,LY,LOUTFL)
      IF (LOUTFL) THEN
         PFFPAR(IPUFF,0)=0.0
         GO TO 1000
      ELSE
         IF (ZZ.LT.RHS(LX,LY,1)) ZZ=RHS(LX,LY,1)
         IF (ZZ.GT.RHS(LX,LY,2)) ZZ=RHS(LX,LY,2)
         CALL ADVECT(XX,YY,ZZ,TRAVEL)
         CALL DEFINE(IPUFF,XX,YY,ZZ,TRAVEL)
         IF(DBUG.AND.MOD(IPUFF,15).EQ.1) PRINT6000,IPUFF,LOUTFL,
    $    (PUFINT(IPUFF,JJP),JJP=0,4),(PFFPAR(IPUFF,KKP),KKP=0,2)
      END IF
6000  FORMAT(' PUFF-STUFF(ADV2ND)--#,OUT?,MAXMIX,X,Y,Z,AGE,Q,VTZ,VTX'
    $        ,1X,I2,L3,6F8.0,2E12.2)
C
C  SET MATERIAL IN PUFF TO 0 FOR FUTURE REMOVAL IF OUT OF BOUNDS
C
      IF (LOUTFL)    PFFPAR(IPUFF, 0)=0
1000  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BASICS (VDISR0,VDISZ0,IHOUR)
C
C     THIS ROUTINE READS IN BASIC INFORMATION AND MAKE SOME
C     REQUIRED CALCULATIONS
C
      LOGICAL DBUG,LOUTFL
     PARAMETER   (NEIGN=9,MXNPF=199,NRCPTX=20,NRCPTY=20)
     COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
    $                DOTX(0:1,0:NEIGN)
     COMMON /SOURCE/ XSOURC,YSOURC,ZSOURC,QSOURC
     COMMON /PEFCON/ PUFINT(0:MXNPF,0:4),PFFPAR(0:MXNPF,0:2),RSPACM,
    $                CONCEN(0:NRCPTX,0:NRCPTY,0:5),GSIGZ(6),
    $                ALPHAZ(6),ALPHAY(6),FEXP(0:90),RCPTRX(0:NRCPTX),
    $                RCPTRY(0:NRCPTY),XRCPO,YRCPO
     COMMON /LOGES/  LOUTFL,DBUG
     DIMENSION VDISR0(6),VDISZ0(6)
     DATA  SIGZ0, SIGY0
    $    /  3.0,   8.0 /
     DO 260 I=0,90
C
C  CALC  EXPS AND INITIAL VIRTUAL TRAVEL
C
            IF (I.GT.6 .OR. I.EQ.0) GO TO 250
            VDISZ0(I)=(SIGZ0/ALPHAZ(I))**(1.0 /GSIGZ(I))
            VDISR0(I)=(SIGY0/ALPHAY(I))**(1.1111)
250         FEXP(I)=EXP (-0.1*I)
260  CONTINUE
C
C  INPUT OF BASIC GRID,WIND STATION INFO ETC
C
     PRINT *, 'RECEPTOR SPACING (M)?'
     ACCEPT *, RSPACM
     PRINT*,' ',  RSPACM
     PRINT *, 'HOW MANY WIND SITES?'
     ACCEPT *, NWINDS
     PRINT*,' ', NWINDS
     PRINT *, 'HOW MANY EMP. ORTH. FCTNS?'
     ACCEPT *,NFUNCT
     PRINT*,' ', NFUNCT
     IF (NFUNCT.LE. 2*(NWINDS +1)) GO TO 340
     STOP 'TOO MANY EIGENVECTORS SPECIFIED'
340  PRINT *, 'SOURCE X,Y,Z (ABOVE SFC) IN METERS?'
     ACCEPT *,XSOURC,YSOURC,ZSOURC
     PRINT*,' ', XSOURC,YSOURC,ZSOURC
     CALL GRIDXY (DBUG)
     PRINT *, 'HOUR?'
     ACCEPT *,IHOUR
     PRINT*,' ', IHOUR
C
C     PUT HR BETWEEN 0 & 24
C
     IHOUR=NEXTHR(IHOUR)
C
C  READ EMPIRICAL ORTHOGONAL FUNCTIONS
C
     CALL EVCO
     RETURN
     END
```

```
      SUBROUTINE CALCON(ITYP,ISTAB,ZZ,TYPMDL,QMULTF,II,JJ,I,
     $                   SIGZ,NHOUR,HEREMX)
C
C  CALCULATES CONCENTRATION CONTRIBUTION (LATERAL EFFECTS ALREADY
C  ACC'NTED FOR IN QMULTF) ACCORDING TO APPROPRIATE VERTICAL MODEL
C  ITYP--1=BOX,2=UNREFLECTED & 3=REFLECTED
C  CONCEN(I,J,K)--I,J=X,Y RECEPTOR INDICES FOR Kth HOUR OF SEQUENCE
C  TO IDENTIFY CONCENTRATION (G/CU.M). NOTE-- ZZ IS RELATIVE TO THE
C  SFC HEIGHT IN THIS ROUTINE.ISTAB=STABILITY CLASS; QMULTF= FACTOR THAT
C  INCLUDES MATERIAL CONTENT &RADIAL SPREADING EFFECTS. II&JJ ARE RECEPTOR
C  GRID INDICES;I=PUFF #;SIGZ=VERTICAL STNDRD DEV.;NHOUR = # OF HRS SINCE
C  START.HEREMX=LOCAL MIXING DEPTH.
C
      LOGICAL DBUG,LOUTFL
      PARAMETER  (NRCPTX=20,NRCPTY=20,MXNPF=199,NEIGN=9)
      COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
     $                DOTX(0:1,0:NEIGN)
      COMMON /PFFCON/ PUFINT(0:MXNPF,0:4),PFFPAR(0:MXNPF,0:2),RSPACM,
     $                CONCEN(0:NRCPTX,0:NRCPTY,0:5),GSIGZ(6),
     $                ALPHAZ(6),ALPHAY(6),FEXP(0:90),RCPTRX(0:NRCPTX),
     $                RCPTRY(0:NRCPTY),XRCPO,YRCPO
      COMMON /LOGES/ LOUTFL,DBUG
      IF  (DBUG .AND. JMOD(I,15).EQ.0) PRINT *, ' IN CALCON'
      IF  (ITYP-2) 2720, 2770, 2810
C
C  BOX MODEL
C
2720       IF(DEPTH.GT.PUFINT(I, 0)) THEN
                 PFFPAR(I,1)=((SQRT(3.8*TYPMDL))/ALPHAZ(ISTAB))
     $                     **(1.0/GSIGZ(ISTAB))
                 PUFINT(I, 0)=HEREMX
           END IF
           CONCEN(II,JJ,NHOUR)=CONCEN(II,JJ,NHOUR)+
     $                        QMULTF*1.25/PUFINT(I, 0)

        GO TO 2870
C
C  NO REFLECTION MODEL
C
2770       K=(5.0 *ZZ*ZZ)/(SIGZ*SIGZ)+0.5
           IF  (K.GT.90) GO TO  2870
           CONCEN(II,JJ,NHOUR)=(QMULTF*FEXP(K)/SIGZ)+CONCEN(II,JJ,NHOUR)
        GO TO 2870
C
C  REFLECTION MODEL
C
2810       QZ=(2.0*HEREMX-ZZ)/SIGZ
           K=QZ*QZ*5.0+0.5
           IF  (K.GT.90) GO TO 2770
           CONCEN(II,JJ,NHOUR)=(QMULTF*FEXP(K)/SIGZ)+CONCEN(II,JJ,NHOUR)
C
C  GO BACK AND GET THE UNREFLECTED PART
C
           GO TO 2770
2870      IF(DBUG) PRINT 6000,I,II,JJ,CONCEN(II,JJ,NHOUR)
6000      FORMAT(' PUFF ',I3,' CUMULATIVE CONCENTRATION IN CELL',
     $            2I3,' IS ',E12.3)
        RETURN
        END
```

```
C
      SUBROUTINE DEFINE(IPUFF,XX,YY,ZZ,TRAVEL)
C
C  DEFINES PARAMETERS FOR EACH PUFF --1ST INDEX IDENTIFIES PUFF
C  2ND INDICES IDENTIFY VARIOUS CHARACTERISTICS AS FOLLOWS:
C                PUFINT--
C                         0=MAX MIX DEPTH ENCOUNTERED (M)
C                         1,2,3=X,Y,Z COORDINATES (M)
C                         4= AGE OF PUFF (MIN.)
C                PFFPAR--
C                         0=MATERIAL IN PUFF (GM)
C                         1,2= VIRT TRAVL DIST. FOR SIGZ &SIGY (M)
C
        LOGICAL LOUTFL,DBUG
      PARAMETER   (MXNPF=199,NRCPTX=20,NRCPTY=20)
      COMMON /PFFCON/ PUFINT(0:MXNPF,0:4),PFFPAR(0:MXNPF,0:2),RSPACM,
     $                CONCEN(0:NRCPTX,0:NRCPTY,0:5),GSIGZ(6),
     $                ALPHAZ(6),ALPHAY(6),FEXP(0:90),RCPTRX(0:NRCPTX),
     $                RCPTRY(0:NRCPTY),XRCPO,YRCPO
      COMMON /LOGES/ LOUTFL,DBUG
       IF (LOUTFL) GO TO  2680
C
C  ASSIGN PUFF PARAMETERS AT END OF 10-MIN STEP
C
       PUFINT(IPUFF,1)=XX
       PUFINT(IPUFF,2)=YY
       PUFINT(IPUFF,3)=ZZ
       PUFINT(IPUFF,4)=PUFINT(IPUFF,4)+10
       PFFPAR(IPUFF,1)=PFFPAR(IPUFF,1)+TRAVEL
       PFFPAR(IPUFF,2)=PFFPAR(IPUFF,2)+TRAVEL
       IF(DBUG .AND.  (MOD(IPUFF,15).EQ.1)) PRINT6000,
     $      IPUFF,(PUFINT(IPUFF,L),L=0,4),
     $      (PFFPAR(IPUFF,L),L=0,2)
6000    FORMAT(1X,'PUFF PARAMS (DEFINE)--#,MAXMIX,X,Y,Z,AGE,Q,
     $VTZ,VTY' ,1X,I5,6F8.0,2E12.2)
2680   RETURN
       END
C
C*******************************************************************
C



        SUBROUTINE DOTINT (ITIME)
C
C       THIS SUB INTERPOLATES BETWEEN INNER PRODUCTS FOR THE BEGINNING
C       AND END OF THE HOUR TO GET VALUES FOR THIS TIME STEP
C
        LOGICAL DBUG
        PARAMETER (NEIGN=9)
      COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
     $                DOTX(0:1,0:NEIGN)
      COMMON /LOGES/  LOUTFL,DBUG
      DO 570 I=1, NFUNCT
         DOTP(I)=DOTX(0,I)+(0.5+ITIME)*(DOTX(1,I)-DOTX(0,I))/6.0
         IF(DBUG) PRINT*,' INNR PROD.' ,I,DOTP(I)
570   CONTINUE
      DOTP(0)=1.0
      RETURN
        END
```

```
C
      SUBROUTINE EVCO
C
     LOGICAL DBUG,LOUTFL
     PARAMETER   (NEIGN=9,NSTA=8)
     COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
    $                DOTX(0:1,0:NEIGN)
     COMMON /WNDEIG/ UM(0:NSTA+1),VM(0:NSTA+1),
    $                UE(0:NEIGN,0:NSTA+1),VE(0:NEIGN,0:NSTA+1)
     COMMON /LOGES/ LOUTFL,DBUG
C
C  READS EIGENVECTORS AND MEAN WINDS,CORRECTS UNITS & REORDERS
C  EV'S SO THAT 1ST ONE EXPLAINS MOST VARIANCE ETC
C
      IVREAD=11
      INVEIG=NEIGN-NFUNCT
      IF (INVEIG.LT.1) STOP 'TOO MANY EIGENVECTS SPECIFIED'
      DO 10 K=0,NFUNCT
C
C  INPUT FILES ORDERED SO THAT:
C       IV=0 IS MEAN
C       IV=NEIGN+1 IS MOST VARIANCE EIGENVECT.
C       IV=NEIGN     " 2ND MOST  "
C       IV=NEIGN-1   " 3RD  "     " ETC ETC
              IF(DBUG) PRINT*, 'EV FILE#',IVREAD
              READ(IVREAD,11) IV
              IF(IV.EQ.0) THEN
                   DO 8 J=0,NWINDS
                        READ(IVREAD,12) UM(J),VM(J)
C----------------------------------------------------------------------
C  CONVERTING CM/SEC TO M/SEC. ENDLICH'S WIND MODELS OUTPUT IN CM/SEC--
C  CODE MODIFICATIONS MAY BE REQUIRED FOR USE WITH OTHER MODELS.
C----------------------------------------------------------------------
                        UM(J)=UM(J)/100.
                        VM(J)=VM(J)/100.
                        IF(DBUG) PRINT*, 'MEAN U,V',
    $                     UM(J),VM(J)
 8                 CONTINUE
              ELSE
                   IV=NEIGN-IV+2
                   DO 9 J=0,NWINDS
                        IF(IV.LT.0) STOP 'IV < 0'
                        READ(IVREAD,12) UE(IV,J),VE(IV,J)
C----------------------------------------------------------------------
C                  CONVERTING CM/SEC TO M/SEC. (SEE ABOVE).
C----------------------------------------------------------------------
                        UE(IV,J)=UE(IV,J)/100.

                        VE(IV,J)=VE(IV,J)/100.
                         IF(DBUG) PRINT*, 'EV #',IV,' U,V',
    $                      UE(IV,J),VE(IV,J)
  9                CONTINUE
              END IF
  10  CONTINUE
  11  FORMAT(I10)
  12  FORMAT(2F10.2)
C
      RETURN
      END
C
```

```
      SUBROUTINE FILEID (IREAD,DEPTH)
C
C     CHECKING FOR APPROPRIATE FILE IDENTIFIER FOR READING
C     APPROPRIATE PRECALCULATED WIND FIELD SOLUTIONS.
      INTEGER DNI
C-----------------------------------------------------------------------
C  PARAMETER NFILES=THE # OF SOLUTION FILES STORED IN LU
C   #'S 21,22,...(20+NFILES)
C-----------------------------------------------------------------------
      PARAMETER (NFILES=2)
      IREAD=0
      TMIN=100000.
C-----------------------------------------------------------------------
C  FIND FILE WITH AVTHK CLOSEST TO MIXING DEPTH; FILES FROM ENDLICH MODELS ARE
C  IN CM--CODE MODIFICATIONS MAY BE REQUIRED FOR USE WITH OTHER MODELS.
C-----------------------------------------------------------------------
      DO 50 IR=21,NFILES+20
            REWIND IR
            READ(IR,9091) JSITE,IV,DNI,AVTHK,SLFAC
            REWIND IR
C
C  NOTE CONVERSION TO METERS.
C
            TEMP=ABS(AVTHK/100.-DEPTH)
            IF(TEMP.LT.TMIN) THEN
                  TMIN=TEMP
                  IREAD=IR
            END IF
   50 CONTINUE
      IF(IREAD.LE.20) STOP 1000
 9091 FORMAT(3I5,F10.0,F5.1)
      RETURN
      END
```

```
      SUBROUTINE GETWND(IX,IY,IZ)
C
C  CALCULATES WIND IN CELL IX,IY,IZ FROM PRECALCULATED EIGENVCTR
C  SOL'NS (0 TO NFUNCT) USING LOG-LINEAR INTERPOLATION (CINTRP(IZ)
C  PROVIDES APPROPRIATE LOG INTERPOLATION FACTORS) BETWEEN
C  VALUES AT TOP & BOTTOM, & INNER PRODUCTS (DOTP) LINEARLY
C  INTERPOLATED IN TIME.
C
      LOGICAL DBUG,LOUTFL
      PARAMETER  (INDX=9,INDY=9,INDZ=5,NEIGN=9)
      DIMENSION CINTRP(0:INDZ)
     COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
    $                DOTX(0:1,0:NEIGN)
     COMMON /WNDFLD/ WNDUVW(0:INDX,0:INDY,0:INDZ,0:2),GSPACE,
    $                PRECAL(0:INDX,0:INDY,0:1,0:2,0:NEIGN),
    $                RHS(0:INDX,0:INDY,2),XORIGN,YORIGN,ZORIGN

     COMMON /LOGES/ LOUTFL,DBUG
      DATA CINTRP /0.24,0.51,0.68,0.80,0.95,0.97/
      FINTRP=CINTRP(IZ)
C
C  SETTING UNCALCULATED COMPONENTS TO 0
C
      DO 50 K=0,2
              WNDUVW(IX,IY,IZ,K)=0.0
 50   CONTINUE
C
C  SUMMING MEAN SOL'N & CONTRIBUTIONS FROM EV SOL'NS, THEN
C  INTERPOLATING TO LEVEL IZ.
C
      DO 100 K=0,NFUNCT
C---------------------------------------------------------------------
C  TAKES MEAN SOL'N (K=0) AND ADDS PRODUCTS OF INNER PRODUCTS (DOTP,DOTP(0)=1)
C  AND SOL'NS FOR VARIOUS EIGENVECTORS TO GET WIND AT TOP & BOTTOM OF LAYER.
C  NOTE, VALUES ARE ALSO CONVERTED FROM CM/SEC TO M/10-MIN.--CODE MODIFICATIONS
C  MAY BE REQUIRED IF ENDLICH'S MODEL IS NOT USED.
C---------------------------------------------------------------------
              UL=PRECAL(IX,IY,0,0,K)*6.
              UU=PRECAL(IX,IY,1,0,K)*6.
              VL=PRECAL(IX,IY,0,1,K)*6.
              VU=PRECAL(IX,IY,1,1,K)*6.
              WL=PRECAL(IX,IY,0,2,K)*6.
              WU=PRECAL(IX,IY,1,2,K)*6.
              WNDUVW(IX,IY,IZ,0)=DOTP(K)*(UL+(UU-UL)*FINTRP)
    $                          +WNDUVW(IX,IY,IZ,0)
              WNDUVW(IX,IY,IZ,1)=DOTP(K)*(VL+(VU-VL)*FINTRP)
    $                          +WNDUVW(IX,IY,IZ,1)
              WNDUVW(IX,IY,IZ,2)=DOTP(K)*(WL+(WU-WL)*FINTRP)
    $                          +WNDUVW(IX,IY,IZ,2)
              IF(DBUG.AND.K.EQ.3) PRINT 9093,IX,IY,IZ,
    $                   (WNDUVW(IX,IY,0,LL),LL=0,2)
 100  CONTINUE
 9093 FORMAT(1X,'U,V&W FROM CELL (GETWND)',3I4,3F10.2)
      RETURN
      END
```

```
C
      SUBROUTINE GRAFF (I,DBUG)
C
C-----------------------------------------------------------------------
C     THIS ROUTINE USES THE NCAR GRAPHICS PKG ROUTINE EZSRFC TO
C     PLOT A 3-D PRESENTATION OF THE CONCENTRATION FIELD.--CODE
C     MODIFICATIONS MAY BE REQUIRED IF OTHER GRAPHING PACKAGES ARE USED.
C-----------------------------------------------------------------------
C
      PARAMETER   (NRCPTX=20,NRCPTY=20,MXNPF=199)
      COMMON /PFFCON/ PUFINT(0:MXNPF,0:4),PFFPAR(0:MXNPF,0:2),RSPACM,
     $                CONCEN(0:NRCPTX,0:NRCPTY,0:5),GSIGZ(6),
     $                ALPHAZ(6),ALPHAY(6),FEXP(0:90),RCPTRX(0:NRCPTX),
     $                RCPTRY(0:NRCPTY),XRCPO,YRCPO
      LOGICAL DBUG
      DIMENSION Z(21,21),WORK(1300)
      DATA ANGH, ANGV
     $     / 60.,   210./
      MX=NRCPTX+1
      MY=NRCPTY+1
      IF (MX*MY.GT.625) STOP
     $          'TOO MANY RECEPTORS FOR GRAPHING'
      DO 92 J=1,MX
      DO 90 K=1,MY
          Z(J,K)=CONCEN(J-1,K-1,I)
 90   CONTINUE
      IF (DBUG) PRINT 6000, (Z(L,J),L=1,21)
6000  FORMAT (1X,11E10.2/5X,10E10.2/)
 92   CONTINUE
      IF (DBUG) GO TO 95
C-----------------------------------------------------------------------
C  USING NCAR GRAPHING ROUTINE-- CONCENTRATION PATTERN IN 3-D
C  VIEWED FROM ANGH DEG. SOUTH OF X AXIS & ANGV DEG. ABOVE HORIZONTAL.
C
C     CALL EZSRFC (Z,MX,MY,ANGH,ANGV,WORK)
C-----------------------------------------------------------------------
C  USING NCAR CONTOUR ROUTINE FOR REGULARLY SPACED, RECTANGULAR MX-BY-MY
C  ARRAY OF VALUES.
      CALL EZCNTR(Z,MX,MY)
C-----------------------------------------------------------------------
 95   RETURN
      END
```

```
      SUBROUTINE GRIDXY(DBUG)
C
C  CALCULATES RECEPTOR COORDINATES WITH GRID CENTERED ON SOURCE
C
        LOGICAL DBUG
      PARAMETER   (NRCPTX=20,NRCPTY=20,MXNPF=199)
      COMMON /PFFCON/ PUFINT(0:MXNPF,0:4),PFFPAR(0:MXNPF,0:2),RSPACM,
     $                CONCEN(0:NRCPTX,0:NRCPTY,0:5),GSIGZ(6),
     $                ALPHAZ(6),ALPHAY(6),FEXP(0:90),RCPTRX(0:NRCPTX),
     $                RCPTRY(0:NRCPTY),XRCPO,YRCPO
      COMMON /SOURCE/ XSOURC,YSOURC,ZSOURC,QSOURC
      XRCPO=XSOURC-((NRCPTX/2.0)*RSPACM)
      YRCPO=YSOURC-((NRCPTY/2.0)*RSPACM)
        IF (DBUG) PRINT *, ' RECPTR ORIG--',XRCPO,YRCPO
      DO 3140 JJ=0 ,NRCPTX
         RCPTRX(JJ)=XRCPO+RSPACM*JJ
         IF (DBUG) PRINT *, ' RCPTR X COORD--',JJ,RCPTRX(JJ)
3140  CONTINUE
      DO 3150 JJ=0 ,NRCPTY
         RCPTRY(JJ)=YRCPO+RSPACM*JJ
         IF (DBUG) PRINT *, ' RCPTR Y COORD--',JJ,RCPTRY(JJ)
3150  CONTINUE
      RETURN
      END
C
C******************************************************************************
C
        SUBROUTINE INDXY(XX,YY,LX,LY,LOUTFL)
C
C  FINDS THE GRID SQUARE INDICES LX & LY FOR THE COORDINATES XX & YY.
C  SETS LOUTFL TO TRUE IF OUT OF BOUNDS.
C
        PARAMETER (INDX=9,INDY=9,INDZ=5,NEIGN=9)
        LOGICAL LOUTFL
      COMMON /WNDFLD/ WNDUVW(0:INDX,0:INDY,0:INDZ,0:2),GSPACE,
     $                PRECAL(0:INDX,0:INDY,0:1,0:2,0:NEIGN),
     $                RHS(0:INDX,0:INDY,2),XORIGN,YORIGN,ZORIGN
          LOUTFL=.TRUE.
          LX=(XX-XORIGN)/GSPACE
          IF (LX.LT.0 .OR. LX.GT.INDX) GO TO  2500
          LY=(YY-YORIGN)/GSPACE
          IF (LY.LT.0 .OR. LY.GT.INDY) GO TO  2500
          LOUTFL=.FALSE.
 2500     RETURN
          END
```

```
C
      SUBROUTINE INFLU (GSIGY,NHOUR,ISTAB)
C
C     THIS SUBROUTINE DETERMINES WHICH RECEPTORS ARE CLOSE ENOUGH TO
C     THIS PUFF TO RECEIVE A SIGNIFICANT CONTRIBUTION TO THE CONCENTRATION,
C     AND THEN ACCUMULATES THE CONTRIBUTIONS AT THE AFFECTED RECEPTORS.
C
      PARAMETER (MXNPF=199,NRCPTX=20,NRCPTY=20,NEIGN=9)
      PARAMETER (INDX=9,INDY=9,INDZ=5)
      LOGICAL DBUG,LOUTFL
     COMMON /WNDFLD/ WNDUVW(0:INDX,0:INDY,0:INDZ,0:2),GSPACE,
    $                PRECAL(0:INDX,0:INDY,0:1,0:2,0:NEIGN),
    $                RHS(0:INDX,0:INDY,2),XORIGN,YORIGN,ZORIGN
     COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
    $                DOTX(0:1,0:NEIGN)
     COMMON /SOURCE/ XSOURC,YSOURC,ZSOURC,QSOURC
     COMMON /PFFCON/ PUFINT(0:MXNPF,0:4),PFFPAR(0:MXNPF,0:2),RSPACM,
    $                CONCEN(0:NRCPTX,0:NRCPTY,0:5),GSIGZ(6),
    $                ALPHAZ(6),ALPHAY(6),FEXP(0:90),RCPTRX(0:NRCPTX),
    $                RCPTRY(0:NRCPTY),XRCPO,YRCPO
     COMMON /LOGES/  LOUTFL,DBUG
      NPNT=-1
     DO 1700 I=0,NOPUFF
C
C  GET HT. ABOVE SFC TO DETERMINE PROPER MODEL TYPE.
C
       CALL INDXY(PUFINT(I,1),PUFINT(I,2),LX,LY,LOUTFL)
       IF (LOUTFL) GO TO 1700
       ZZ=PUFINT(I, 3)-RHS(LX,LY,1)
       IF(DBUG.AND. MOD(I,15).EQ.0)THEN
           PRINT*,'(IN SUBROUTINE INFLU)-#,TYPMDL,HEREMX,X,Y,Z'
           PRINT *,I,TYPMDL,HEREMX,(PUFINT(I,JK),JK=1,2),ZZ
       END IF
C
C  GET LOCAL DEPTH OF LAYER
C
       HEREMX=RHS(LX,LY,2)-RHS(LX,LY,1)
       TYPMDL=HEREMX*(HEREMX-ZZ)
       IF (TYPMDL.LT. 0) GO TO  1700
C
C  CHECK FOR ABOVE MIX DEPTH
C
       SIGZ=(ALPHAZ(ISTAB)*(PFFPAR(I, 1)**GSIGZ(ISTAB)))
       BIGTOP=SQRT (TYPMDL/0.26)
       IF (SIGZ.GT.BIGTOP)  SIGZ=BIGTOP+1
       IF (PUFINT(I, 3).GT.3.0*SIGZ) GO TO  1700
       SYX3=(ALPHAY(ISTAB)*PFFPAR(I, 2)**GSIGY)*3.0
C
C               SET SYX3=3SIGMA & FIND BOUNDS OF PUFF INFLUENCE
C
       LOWCOL=NINT((PUFINT(I,1)-XRCPO-SYX3)/RSPACM)-1
       IF (LOWCOL.GT.NRCPTX) GO TO  1700
       MAXCOL=NINT((PUFINT(I,1)-XRCPO+SYX3)/RSPACM)+1
       IF (MAXCOL.LT. 0) GO TO  1700
       LOWROW=NINT((PUFINT(I,2)-YRCPO-SYX3)/RSPACM)-1
       IF (LOWROW.GT.NRCPTY) GO TO  1700
       MAXROW=NINT((PUFINT(I,2)-YRCPO+SYX3)/RSPACM)+1
       IF (MAXROW.LT. 0) GO TO  1700
       IF (MAXCOL.GT.NRCPTX)  MAXCOL=NRCPTX
       IF (LOWCOL.LT. 0)  LOWCOL=0
       IF (MAXROW.GT.NRCPTY)  MAXROW=NRCPTY
       IF (LOWROW.LT. 0)  LOWROW=0
C
C  RESET SIGY TO 1/3 THE RANGE OF INFLUENCE
C
       SIGY=SYX3/3.0
```

```
C
C  SELECT MODEL TYPE--BOX MODEL (ITYP=1), NO REFLECTION (ITYP=2),
C                     REFLECTION (ITYP=3).
C
        ITYP=2
        IF (DBUG .AND. MOD(I,15).EQ.0) PRINT*,' LCL DPTH,TYPMDL,SIGZ',
     $          HEREMX,TYPMDL,SIGZ
        IF (DEPTH.LE. 0) GO TO  1610
        IF ((1.15*SIGZ*SIGZ).LE.TYPMDL) GO TO  1610
        ITYP=3
        IF ((0.26*SIGZ*SIGZ).LE.TYPMDL) GO TO  1610
        ITYP=1
1610    IF(DBUG .AND. MOD(I,15).EQ.0) THEN
                PRINT*, ' #,LOWC,MAXC,LOWR,MAXR ',
     $                   I,LOWCOL,MAXCOL,LOWROW,MAXROW
                PRINT*, ' SIGZ,SIGY ',SIGZ,SIGY
        END IF
C
C  NOW CALCULATE CONCENTRATION FOR AFFECTED REGION.
C
        DO 1690 II=LOWCOL,MAXCOL
          DO 1690 JJ=LOWROW,MAXROW
            NPNT=NPNT+1
            XX=PUFINT(I, 1)-RCPTRX(II)
            YY=PUFINT(I, 2)-RCPTRY(JJ)
C
C  GET HT. OF PUFF ABOVE RECEPTOR TO CALCULATE CONCEN.
C
            CALL INDXY(RCPTRX(II),RCPTRY(JJ),LX,LY,LOUTFL)
            ZZ=PUFINT(I,3)-RHS(LX,LY,1)
            INDEXP=(5.0*(XX*XX+YY*YY)/(SIGY*SIGY))
            IF (DBUG .AND. MOD(NPNT,15).EQ. 0) PRINT *,
     $      ' ROW, COL,X,Y,Z, RCPTRX, RCPTRY, SIGY, INDEXP',
     $      II,JJ,XX,YY,ZZ,RCPTRX(II),RCPTRY(JJ),SIGY,INDEXP
            IF (INDEXP.GT.90) GO TO  1690
C
C  GET FACTOR (QMULTF) THAT INCLUDES PUFF MATERIAL & LATERAL SPREADING
C  EFFECTS & CALL CALCON TO INCREMENT CONCENTRATIONS AT AFFECTED RECEPTOR
C
            QMULTF=(2*PFFPAR(I, 0)*FEXP(INDEXP))
     $              /(15.75*SIGY*SIGY)
            CALL CALCON(ITYP,ISTAB,ZZ,TYPMDL,QMULTF,
     $                  II,JJ,I,SIGZ,NHOUR,HEREMX)
1690     CONTINUE
1700   CONTINUE
       RETURN
       END
```

```
      SUBROUTINE INWIND(I,IHOUR)
C
C  THIS SUB ACCEPTS WIND, CALCS COMPONENTS, FORMS INNER PRODUCTS
C  WITH EIGENVECTORS (OR OTHER ORTHOG.FNCTNS)
C
        LOGICAL DBUG,LOUTFL
      PARAMETER   (NEIGN=9,NSTA=8)
      COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
     $                DOTX(0:1,0:NEIGN)
        COMMON /WNDEIG/ UM(0:NSTA+1),VM(0:NSTA+1),
     $                UE(0:NEIGN,0:NSTA+1),VE(0:NEIGN,0:NSTA+1)
      COMMON /LOGES/ LOUTFL,DBUG
      DIMENSION U(0:NSTA+1),V(0:NSTA+1)
C
C  CORRECTING FOR 1ST HOUR INPUTS
C
       IF (I.EQ.0)  IHOUR=IHOUR-1
C
C             INPUT WINDS & GET COMPS
C
       IF (I.LE. 0) GO TO  2200
       DO 2190 J=1,NEIGN
          DOTX(0,J)=DOTX(1 ,J)
2190   CONTINUE
2200   DO 2250 J=0,NWINDS
         IF(J.EQ. NWINDS) THEN
              PRINT *, 'GRADIENT WIND SPD. (M/S) & DIR.
     $                FOR HR.',IHOUR,'?'
         ELSE
              PRINT *, 'WIND SPD(M/S) & DIR. AT SITE ',
     $                 J+1 ,' FOR HR ',IHOUR
         END IF
          ACCEPT *, WNDSPD,WD
          PRINT*,' ',WNDSPD,WD
C
C  CONVERTING TO DEVIATIONS FROM MEAN COMPONENTS
C
          U(J)=WNDSPD*COS(WINDIR(WD))-UM(J)
          V(J)=WNDSPD*SIN(WINDIR(WD))-VM(J)
          IF(DBUG) PRINT*,' U,V AT SITE # ',J,U(J),V(J)
2250   CONTINUE
C
C  ZERO INNER PRODS & GET DOTX'S--INDEX I=HR & K=FUNCTION
C
       DO 2275 K=1,NFUNCT
          DOTX(I,K)=0.0
2275   CONTINUE
       DO 2300 K=1,NFUNCT
          DO 2290 J=0,NWINDS
            DOTX(I,K)=U(J)*UE(K,J)+V(J)*VE(K,J)+DOTX(I,K)
2290      CONTINUE
          IF(DBUG) PRINT*, 'INNR PRD #',K,DOTX(I,K),I
2300   CONTINUE
C
C  RESET HOUR AFTER 1ST HOUR INPUTS
C
       IF (NOPUFF.EQ.0)THEN
          DO 2305 K=1,NFUNCT
              DOTX(1,K)=DOTX(0,K)
2305      CONTINUE
          IHOUR=NEXTHR(IHOUR+1)
          NOPUFF=NOPUFF+1
       END IF
       RETURN
       END
```

```
        SUBROUTINE MET IN (QUIT,ISTAB,IHOUR,ALPHAY,SEPMAX)
C
C       THIS ROUTINE READS IN BASIC METEOROLOGICAL INFORMATION FOR THE
C       NEXT HOUR & SETS QUIT FLAG TO STOP COMPUTATIONS IF A NEGATIVE
C       MIXING DEPTH IS INPUT.
C
        LOGICAL QUIT
      PARAMETER  (NEIGN=9)
      DIMENSION ALPHAY(6)
      COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
     $                DOTX(0:1,0:NEIGN)
      COMMON /SOURCE/ XSOURC,YSOURC,ZSOURC,QSOURC
      QUIT=.FALSE.
      PRINT *, 'MIX HT. (M) & STABIL.?--NEG. MIX HT TO STOP'
      ACCEPT *,DEPTH,ISTAB
      PRINT*,' ', DEPTH,ISTAB
       IF (DEPTH.LE. 0) THEN
           QUIT=.TRUE.
       ELSE
           SEPMAX=ALPHAY(ISTAB)*0.45
           PRINT *, 'SOURCE STRNGTH (G/S)?'
           ACCEPT *,QSOURC
           PRINT*,' ', QSOURC
           IHOUR=NEXTHR(IHOUR+1)
C
C  FIRST HOUR WINDS ARE INPUT & INNER PRODUCTS CALCULATED
C
           IF (NOPUFF.LE. 0) THEN
              I=0
              CALL INWIND(I,IHOUR)
           END IF
C
C  INPUT NEXT HOUR'S WIND & CALC. INNR PROD.
C
           I=1
           CALL INWIND(I,IHOUR)
       END IF
        RETURN
        END
```

```
      SUBROUTINE MRG PRG (LASTPF,SEPMAX)
C
C     THIS SUBROUTINE MERGES CLOSE-TOGETHER PUFFS & ELIMINATES EXTRAS.
C     OUT-OF-BOUNDS PUFFS (IDENTIFIED BY 0 CONTENT) ARE ALSO ELIMINATED.
C
      LOGICAL LOUTFL
      PARAMETER (NEIGN=9,MXNPF=199,NRCPTX=20,NRCPTY=20)
      PARAMETER (INDZ=5,INDX=9,INDY=9)
     COMMON /WNDFLD/ WNDUVW(0:INDX,0:INDY,0:INDZ,0:2),GSPACE,
    $                PRECAL(0:INDX,0:INDY,0:1,0:2,0:NEIGN),
    $                RHS(0:INDX,0:INDY,2),XORIGN,YORIGN,ZORIGN
     COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
    $                DOTX(0:1,0:NEIGN)
     COMMON /PFFCON/ PUFINT(0:MXNPF,0:4),PFFPAR(0:MXNPF,0:2),RSPACM,
    $                CONCEN(0:NRCPTX,0:NRCPTY,0:5),GSIGZ(6),
    $                ALPHAZ(6),ALPHAY(6),FEXP(0:90),RCPTRX(0:NRCPTX),
    $                RCPTRY(0:NRCPTY),XRCPO,YRCPO
      I=0
1015  II=I+1
      IF (I.GE.MXNPF) STOP ' TOO MANY TIMES IN PURGE LOOP'
C
C  SKIP EMPTY PUFFS--CHECK OTHERS & MERGE CLOSE PAIRS
C
      IF(PFFPAR(I,0).EQ. 0) GO TO 1120
      PSEPN=ABS (PFFPAR(II, 2)-PFFPAR(I, 2))
     IF (PSEPN.GE.SEPMAX*PFFPAR(II, 2)) GO TO  1120
     DO 1060 L=0 ,4
       PUFINT(II,L)=0.5*(PUFINT(II,L)+PUFINT(I,L))
1060 CONTINUE
C
C  MAKE SURE MERGED PUFF ISN'T BELOW SFC.
C
       CALL INDXY(PUFINT(II,1),PUFINT(II,2),LX,LY,LOUTFL)
       IF (LOUTFL) THEN
         PFFPAR(II,0)=0.0
         GO TO 1120
       END IF
       IF (PUFINT(II,3).LT.RHS(LX,LY,1)) PUFINT(II,3)=RHS(LX,LY,1)
       IF (PUFINT(II,3).GT.RHS(LX,LY,2)) PUFINT(II,3)=RHS(LX,LY,2)
       PFFPAR(II, 0)=PFFPAR(II, 0)+PFFPAR(I, 0)
       PFFPAR(II, 1)=(PFFPAR(II, 1)+PFFPAR(I, 1))*0.5
       PFFPAR(II, 2)=(PFFPAR(II, 2)+PFFPAR(I, 2))*0.5
       PFFPAR(I, 0)=0
       IF (I.GE.(LASTPF-3)) GO TO  1130
       I=I+2
       GO TO 1015
1120   IF (I.GE.(LASTPF-2)) GO TO  1130
       I=II
       GO TO 1015
1130   IDNEW=-1
       DO 1250 I=0,LASTPF
         IF (PFFPAR(I, 0).LE. 0) GO TO  1250
         IF (PUFINT(I, 3).GT.DEPTH)  PUFINT(I, 3)=DEPTH
C
C  MAXIMUM MIXING DEPTH EFFECT
C
         IF (PUFINT(I, 3).LT. 0)  PUFINT(I, 3)=0
         IDNEW=IDNEW+1
         DO 1240 L=0, 4
           PUFINT(IDNEW,L)=PUFINT(I,L)
           IF (L.GT. 2) GO TO  1240
           PFFPAR(IDNEW,L)=PFFPAR(I,L)
1240    CONTINUE
1250  CONTINUE
      NOPUFF=IDNEW
       RETURN
       END
```

```
      FUNCTION NEXTHR (IHOUR)
C
C  CHECKS TO MAKE SURE IHOUR IS BETWEEN 0 & 23.
C
       NEXTHR=IHOUR
       IF (IHOUR.LT.0)    NEXTHR=IHOUR+24
       IF (IHOUR.GT.23)   NEXTHR=IHOUR-24
       RETURN
       END
C
C**************************************************************************
C
      SUBROUTINE OUTNOW(NHOUR,ISTAB)
C
C  PRINTS SOME BASIC INFO & NON-ZERO CONCENTRATIONS
C
        LOGICAL DBUG,LOUTFL
      PARAMETER   (NRCPTX=20,NRCPTY=20,MXNPF=199,NEIGN=9)
      COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
     $                DOTX(0:1,0:NEIGN)
      COMMON /PFFCON/ PUFINT(0:MXNPF,0:4),PFFPAR(0:MXNPF,0:2),RSPACM,
     $                CONCEN(0:NRCPTX,0:NRCPTY,0:5),GSIGZ(6),
     $                ALPHAZ(6),ALPHAY(6),FEXP(0:90),RCPTRX(0:NRCPTX),
     $                RCPTRY(0:NRCPTY),XRCPO,YRCPO
      COMMON /LOGES/ LOUTFL,DBUG
      COMMON /SOURCE/ XSOURC,YSOURC,ZSOURC,QSOURC
       IF(DBUG) PRINT*, 'STABILITY=', ISTAB
       IF(DBUG) PRINT*, 'MIX DEPTH=', DEPTH
       IF(DBUG) PRINT*, 'SRC HEIGHT=', ZSOURC
       IF(DBUG) PRINT*, 'SRC STRNGTH=', QSOURC
C
C       OUTPUT NON-ZERO CONCENTRATIONS
C
      DO 3350 IHOUR=0,NHOUR-1
         PRINT *,'HOUR NO. ',IHOUR
         DO 3350 II=0 ,NRCPTX
           DO 3350 JJ=0 ,NRCPTY
              IF (CONCEN(II,JJ,IHOUR).EQ.0.0) GO TO  3350
              PRINT *, RCPTRX(II),RCPTRY(JJ),CONCEN(II,JJ,IHOUR)
C
C------------------TO WRITE OUTPUTS TO LOGICAL UNIT 15--------------
C       WRITE (15,6002) IHOUR,RCPTRX(II),RCPTRY(JJ),CONCEN(II,JJ,IHOUR)
C6002   FORMAT (I5,3E15.3)
C-----------------------------------------------------------------
C
3350   CONTINUE
       IF (.NOT. DBUG) GO TO 3357
         PRINT 6000
         DO 3353 I=0,NOPUFF
            SIGY=ALPHAY(ISTAB)*(PFFPAR(I,2)**0.9)
            SIGZ=ALPHAZ(ISTAB)*(PFFPAR(I,1)**GSIGZ(ISTAB))
            PRINT 6001, I, (PUFINT(I,J),J=0,4),PFFPAR(I,0),
     $                  SIGZ,SIGY
3353     CONTINUE
6000     FORMAT ('0 PUFF ID',3X,'MAXMIX',7X,'X',9X,'Y',8X,'Z',
     $           8X,'AGE',5X,'AMOUNT',5X,'SIGZ',6X,'SIGY')
6001     FORMAT (5X,I3,1X,8F10.0)
3357   CONTINUE
       RETURN
       END
```

```
      SUBROUTINE RELHT(RHS,XRHS)
C
C-----------------------------------------------------------------------
C  CALCULATES HEIGHT ABOVE LOWEST PT.  XRHS= ACTUAL HTS. RHS=RELATIVE HTS.
C  FILES FROM ENDLICH MODELS ARE IN CM--CODE MODIFICATIONS MAY BE REQUIRED FOR
C  USE WITH OTHER MODELS.
C-----------------------------------------------------------------------
C
      PARAMETER   (INDX=9,INDY=9)
      DIMENSION XRHS(0:INDX,0:INDY,2),RHS(0:INDX,0:INDY,2)
      ZMIN=1.E9
C
C  FIND LOWEST HT.
C
      DO 50 IX=0,INDX
        DO 50 IY=0,INDY
              ZMIN=AMIN1(ZMIN,XRHS(IX,IY,1))
 50   CONTINUE
C
C  GET RELATIVE HTS & CONVERT TO METERS  (FROM CM)
C
      DO 100 IX=0,INDX
        DO 100 IY=0,INDY
              RHS(IX,IY,1)=(XRHS(IX,IY,1)-ZMIN)/100.0
              RHS(IX,IY,2)=(XRHS(IX,IY,2)-ZMIN)/100.0
              IF(RHS(IX,IY,2).LE.RHS(IX,IY,1)) STOP 'TOP UNDER SFC'
 100  CONTINUE
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE VIRTVL(ISTAB,SL)
C
C        CALCULATES VIRTUAL TRAVEL DISTANCES
C
      INTEGER SL
      PARAMETER   (MXNPF=199,GSYINV=1.0/0.9,NEIGN=9,NRCPTY=20,NRCPTX=20)
      COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
     $                DOTX(0:1,0:NEIGN)
      COMMON /PFFCON/ PUFINT(0:MXNPF,0:4),PFFPAR(0:MXNPF,0:2),RSPACM,
     $                CONCEN(0:NRCPTX,0:NRCPTY,0:5),GSIGZ(6),
     $                ALPHAZ(6),ALPHAY(6),FEXP(0:90),RCPTRX(0:NRCPTX),
     $                RCPTRY(0:NRCPTY),XRCPO,YRCPO
      IF (SL .LT. 1) GO TO  2970
      GSZINV=1.0/GSIGZ(ISTAB)
      DO 2950 LL=0,NOPUFF
C
C  GET SIGMAS FOR PAST STABILITY CLASS  (SL)
C
        SIGZ=ALPHAZ(SL)*(PFFPAR(LL, 1)**GSIGZ(SL))
        SIGY=ALPHAY(SL)*(PFFPAR(LL, 2)**0.9)
C
C  USE SIGMAS TO CALCULATE VIRT TRVL  (M)  FOR NEW STAB. CLASS  (ISTAB)
C
        PFFPAR(LL, 1)=(SIGZ/ALPHAZ(ISTAB))**GSZINV
        PFFPAR(LL, 2)=(SIGY/ALPHAY(ISTAB))**GSYINV
2950  CONTINUE
C
C  SAVE LAST STABILITY CLASS
C
      SL=ISTAB
2970  RETURN
      END
```

```
      SUBROUTINE WIND(XX,YY,ZZ,U1,U2,U3)
C
C  CHECKS FOR OUT OF BOUNDS PUFF & FOR PUFF IN CELL WITH PREVIOUSLY
C  CALCULATED WIND--CALCULATES WIND IF NEEDED.
C
        LOGICAL LOUTFL,DBUG,FIRST
      PARAMETER   (INDX=9,INDY=9,INDZ=5,NEIGN=9)
      COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
     $                DOTX(0:1,0:NEIGN)
      COMMON /WNDFLD/ WNDUVW(0:INDX,0:INDY,0:INDZ,0:2),GSPACE,
     $                PRECAL(0:INDX,0:INDY,0:1,0:2,0:NEIGN),
     $                RHS(0:INDX,0:INDY,2),XORIGN,YORIGN,ZORIGN
      COMMON /LOGES/ LOUTFL,DBUG
C
C  SET OUT OF BOUNDS FLAG & GET HT RELATIVE TO SFC.
C
          CALL INDXY(XX,YY,LX,LY,LOUTFL)
          IF (LOUTFL) GO TO 2500
          LZ=6.0*(ZZ-RHS(LX,LY,1))/(RHS(LX,LY,2)-RHS(LX,LY,1))
          IF (LZ.GT.5) LZ=5
          IF (LZ.LT.0) LZ=0
      FIRST=.FALSE.
C
C   CHECK FOR PRIOR CALCULATION
C
        IF (WNDUVW(LX,LY,LZ,2).EQ.YORIGN) THEN
          CALL GETWND(LX,LY,LZ)
          FIRST=.TRUE.
        END IF
C
C     WIND UNITS ARE M/10-MINUTES
C
        U1=WNDUVW(LX,LY,LZ, 0)
        U2=WNDUVW(LX,LY,LZ, 1)
        U3=WNDUVW(LX,LY,LZ, 2)
         IF(DBUG .AND. FIRST) PRINT 6000, LX,LY,LZ,U1,U2,U3
 6000     FORMAT(1X,'U,V&W FROM CELL (WIND)',3I4,3F10.2)
 2500    RETURN
        END
```

```
      SUBROUTINE WINDDO
            PARAMETER   (INDX=9,INDY=9,INDZ=5,NEIGN=9)
C
C   IDENTIFIES FILE (IREAD) ACCORDING TO MIXING DEPTH (DEPTH)
C   & READS APPROPRIATE SET OF PRECALCULATED WIND FIELD SOL'NS
C
C       IX,IY,IZ =LOCATION GIVEN
C       HTPUFF =HEIGHT OF PUFF IN METERS
C       DEPTH =MIXING DEPTH IN METERS
C       NFUNCT =NO. OF EIGENVECTORS TO BE USED.
C       NYLRS = NO. OF LAYERS, STARTING WITH THE GROUND=1
C       NSTA = NO. OF STATIONS HAVING WIND DATA
C       NTYPE = NO. OF TYPES OF SOLUTIONS e.g. DAY/NITE
C       INDX,INDY,INDZ = DIMENSIONS OF GRID SIZE USED IN X,Y AND Z DIRECTIONS
C       NFILES = NUMBER OF FILES TO READ
C       XRHS = GEOMETRIC HEIGHT ABOVE TERRAIN
C       RHS = GEOMETRIC HT ABOVE THE LOWEST PT IN THE DOMAIN.
C--------------------------------------------------------------------------
C       INPUTS FROM ENDLICH'S MODELS ARE IN CM/SEC AND NEED TO
C       BE CONVERTED TO METERS/SEC.--CODE MODIFICATIONS MAY BE REQUIRED IF
C       OTHER MODELS ARE USED.
C--------------------------------------------------------------------------
C
      DIMENSION XRHS(0:INDX,0:INDY,2)
      COMMON /WINDOB/ NOPUFF,NFUNCT,DEPTH,NWINDS,DOTP(0:NEIGN),
     $                DOTX(0:1,0:NEIGN)
      COMMON /WNDFLD/ WNDUVW(0:INDX,0:INDY,0:INDZ,0:2),GSPACE,
     $                PRECAL(0:INDX,0:INDY,0:1,0:2,0:NEIGN),
     $                RHS(0:INDX,0:INDY,2),XORIGN,YORIGN,ZORIGN
      COMMON /LOGES/ LOUTFL,DBUG
        LOGICAL DBUG,LOUTFL
        INTEGER DNI
        DATA LAST/0/
C
C  NFUNCT= NO. OF EIGENVECTORS TO BE USED
C  GETTING ID OF FILE THAT CORRESPONDS MOST CLOSELY TO MIXING DPTH
C  AND THEN READING SOL'NS IF DEPTH CLASS CHANGED IN PAST HR.
C
        CALL FILEID (IREAD,DEPTH)
        IF (LAST.EQ.IREAD) GO TO 200
        DO 100 K=0,NFUNCT
                READ(IREAD,9091) JSITE,IV,DNI,AVTHK,SLFAC
C
C  IV=K=0 FOR MEAN
C  IV=NEIGN+1 OR K=1 FOR MOST VARIANCE
C  IV=NEIGN OR K=2 FOR 2ND "     "
C  ETC ETC
C
                IF(DBUG) PRINT*,'PRECAL ETC FROM WINDDO'
                IF(DBUG) PRINT*,' JSITE,IV,DNI,AVTHK,SLFAC',K
                IF(DBUG) PRINT 9091,JSITE,IV,DNI,AVTHK,SLFAC
                READ(IREAD,9092) (((PRECAL(I,J,0,0,K),
     $                             PRECAL(I,J,0,1,K),
     $                             PRECAL(I,J,0,2,K),
     $                  XRHS(I,J,1)),J=0,INDY),I=0,INDX)
                IF(DBUG) PRINT 9093,(((PRECAL(I,J,0,0,K),
     $                  PRECAL(I,J,0,1,K)),J=0,INDY),I=0,INDX)
                READ(IREAD,9092) (((PRECAL(I,J,1,0,K),
     $                             PRECAL(I,J,1,1,K),
     $                             PRECAL(I,J,1,2,K),
```

```
     $                   XRHS(I,J,2)),J=0,INDY),I=0,INDX)
               IF(DBUG) PRINT 9093,(((PRECAL(I,J,0,0,K),
     $                 PRECAL(I,J,0,1,K)),J=0,INDY),I=0,INDX)
 100    CONTINUE
        IF (LAST.EQ.0) CALL RELHT(RHS,XRHS)
 200    LAST=IREAD
 9091   FORMAT(1X,I4,2I5,F10.0,F5.1)
 9092   FORMAT(8F10.2)
 9093   FORMAT(1X,10(2F5.0,2X))
        RETURN
        END
C
C**********************************************************************
C
      FUNCTION WINDIR(WD)
      PARAMETER(DEG2R=0.0174533)
C
C  CONVERTS WIND DIRECTION IN METEOROLOGICAL CONVENTION TO MATH
C  ANGLE NOTATION IN RADIANS
C
      WINDIR=(270.-WD)*DEG2R
        RETURN
        END
C
C**********************************************************************
C
      SUBROUTINE WNDBAR(WNDUVW,YORIGN)
      PARAMETER  (INDX=9,INDY=9,INDZ=5,NEIGN=9)
      DIMENSION  WNDUVW(0:INDX,0:INDY,0:INDZ,0:2)
C
C  W COMPONENT SET TO YORIGN IN ORDER TO IDENTIFY CELLS
C  FOR WHICH WIND HAS NOT YET BEEN CALCULATED.
C
      DO 3050  I=0,INDX
        DO 3050  J=0,INDY
            DO 3050  K=0,INDZ
                WNDUVW(I,J,K,2)=YORIGN
3050  CONTINUE
      RETURN
      END
```

Appendix C

SAMPLE PROBLEMS

1. Sample Problem with Spatially Uniform Winds

This example uses artifical eigenvector and wind field solutions that generate winds that are uniform in space, but can change with time. The following eigenvector file is read from logical unit 11:

EIGENVECTOR INPUTS FOR SAMPLE PROBLEM 1

```
       0
    0.00        0.00
    0.00        0.00
    0.00        0.00
      10
    0.00        100.
    0.00        0.00
    0.00        0.00
       9
    100.        0.00
    0.00        0.00
    0.00        0.00
```

The first group of numbers causes the mean u's and v's at the 3 locations (two stations and an upper level wind) to be zero. The next set of numbers will (when used with the solution sets given below) cause the u and v components from the first station to be used throughout the three dimensional field. The corresponding solution sets are shown below. The first file contains dummy solutions for a 1000 meter deep layer (100000 cm); it will be read from logical unit 21. The second file is for a 2000 m deep layer and is read from unit 22.

The interactive inputs are shown after the two solution files. The outputs from this sample run follow. Figure 1 shows graphical outputs obtaining from an NCAR routine.

2. Sample Problem Using Spatially Varying Wind Fields

This example is based on some results provided by Endlich. The input eigenvectors were obtained from data collected in the Goodenoe Hills region of the State of Washington. The first set of solutions is for an average mixing depth of about 600 m. The results were obtained from runs of a mass-conserving model. The second set of results has been synthesized from the first by simply increasing the mixing depth. The input files, the interactive inputs and the outputs for this example are given on the following pages. Figure 2 shows the graphical output.

EIGENVECTOR INPUTS FOR SAMPLE PROBLEM 2

0	
231.34	68.89
37.39	72.13
220.68	-147.67
174.68	-24.78
1027.02	-530.07
10	
-20.61	-5.43
-7.41	-4.91
-14.22	6.77
-5.21	0.47
-93.51	21.15
9	
34.02	-15.01
-0.10	-22.52
19.54	-8.31
16.46	-17.55
-27.96	-79.37
8	
22.65	-22.08
46.91	-50.71
39.16	-27.88
-3.64	-7.96
-2.89	42.80
7	
80.61	-15.48
-27.14	32.29
-7.77	0.91
-28.79	3.24
-8.43	22.63

FIRST WIND SOLUTION FILE (LU=21) FOR SAMPLE PROBLEM 1

99	0	9	100000.	0.9			
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.

0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.
0.00	0.00	0.00	100000.	0.00	0.00	0.00	100000.

99 2 9 100000. 0.9

0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00

0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.

```
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
 00.00    100.00     0.00  100000.     00.00   100.00     0.00  100000.
99     1     9   100000.   0.9
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
100.00      0.00     0.00      0.00   100.00     0.00     0.00      0.00
```

0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	[illegible]	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.00
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.
00.00	100.00	0.00	100000.	00.00	100.00	0.00	100000.

SECOND WIND SOLUTION FILE (LU=22) FOR SAMPLE PROBLEM 1

```
99     0      9    200000.   0.9
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.0
      0.00        0.00        0.00     200000.        0.00        0.00        0.00    200000.
      0.00        0.00        0.00     200000.        0.00        0.00        0.00    200000.
      0.00        0.00        0.00     200000.        0.00        0.00        0.00    200000.
      0.00        0.00        0.00     200000.        0.00        0.00        0.00    200000.
      0.00        0.00        0.00     200000.        0.00        0.00        0.00    200000.
      0.00        0.00        0.00     200000.        0.00        0.00        0.00    200000.
      0.00        0.00        0.00     200000.        0.00        0.00        0.00    200000.
      0.00        0.00        0.00     200000.        0.00        0.00        0.00    200000.
      0.00        0.00        0.00     200000.        0.00        0.00        0.00    200000.
      0.00        0.00        0.00     200000.        0.00        0.00        0.00    200000.
```

0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.
0.00	0.00	0.00	200000.	0.00	0.00	0.00	200000.

99 2 9 200000. 0.9

0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0

0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
0.00	100.00	0.00	0.00	0.00	100.00	0.00	0.0
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.
00.00	100.00	0.00	200000.	00.00	100.00	0.00	200000.

```
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
   00.00    100.00    0.00   200000.    00.00   100.00    0.00  200000.
 99    1    9   200000.   0.9
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
  100.00      0.00    0.00      0.00   100.00     0.00    0.00      0.0
```

100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.0
100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.0
100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.0
100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.0
100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.0
100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.0
100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.0
100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.0
100.00	0.00	0.00	0.00	100.00	0.00	0.00	0.0
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.
100.00	0.00	0.00	200000.	100.00	0.00	0.00	200000.

INTERACTIVE INPUTS FOR PROBLEM 1

```
RECEPTOR SPACING (M)?
    300.0000
HOW MANY WIND SITES?
           2
HOW MANY EMP. ORTH. FCTNS?
           2
SOURCE X,Y,Z (ABOVE SFC) IN METERS?
   0.0000000E+00  0.0000000E+00   30.00000
HOUR?
           7
FOR HR FROM             7 TO            8
MIX HT. (M) & STABIL.?--NEG. MIX HT TO STOP
    900.0000            2
SOURCE STRNGTH (G/S)?
    1000.000
WIND SPD(M/S) & DIR. AT SITE           1 FOR HR               7
    1.000000        270.0000
WIND SPD(M/S) & DIR. AT SITE           2 FOR HR               7
    1.000000        270.0000
GRADIENT WIND SPD. (M/S) & DIR.                  FOR HR.              7?
    1.000000        270.0000
WIND SPD(M/S) & DIR. AT SITE           1 FOR HR               8
    1.000000        270.0000
WIND SPD(M/S) & DIR. AT SITE           2 FOR HR               8
    1.000000        270.0000
GRADIENT WIND SPD. (M/S) & DIR.                  FOR HR.              8?
    1.000000        270.0000
FOR HR FROM             8 TO            9
MIX HT. (M) & STABIL.?--NEG. MIX HT TO STOP
    1800.000            4
SOURCE STRNGTH (G/S)?
    200.0000
WIND SPD(M/S) & DIR. AT SITE           1 FOR HR               9
    2.000000        315.0000
WIND SPD(M/S) & DIR. AT SITE           2 FOR HR               9
    2.000000        315.0000
GRADIENT WIND SPD. (M/S) & DIR.                  FOR HR.              9?
    2.000000        315.0000
FOR HR FROM             9 TO           10
MIX HT. (M) & STABIL.?--NEG. MIX HT TO STOP
    2200.000            4
SOURCE STRNGTH (G/S)?
    200.0000
WIND SPD(M/S) & DIR. AT SITE           1 FOR HR              10
    2.000000        315.0000
WIND SPD(M/S) & DIR. AT SITE           2 FOR HR              10
    2.000000        315.0000
GRADIENT WIND SPD. (M/S) & DIR.                  FOR HR.             10?
    2.000000        315.0000
FOR HR FROM            10 TO           11
MIX HT. (M) & STABIL.?--NEG. MIX HT TO STOP
    500.0000            5
SOURCE STRNGTH (G/S)?
    300.0000
WIND SPD(M/S) & DIR. AT SITE           1 FOR HR              11
    3.000000        45.00000
WIND SPD(M/S) & DIR. AT SITE           2 FOR HR              11
    3.000000        45.00000
GRADIENT WIND SPD. (M/S) & DIR.                  FOR HR.             11?
    3.000000        45.00000
FOR HR FROM            11 TO           12
```

```
MIX HT. (M) & STABIL.?--NEG. MIX HT TO STOP
    500.0000          6
SOURCE STRNGTH (G/S)?
    300.0000
WIND SPD(M/S) & DIR. AT SITE            1 FOR HR            12
    3.000000       45.00000
WIND SPD(M/S) & DIR. AT SITE            2 FOR HR            12
    3.000000       45.00000
GRADIENT WIND SPD. (M/S) & DIR.                  FOR HR.            12?
    3.000000       45.00000
FOR HR FROM            12 TO            13
MIX HT. (M) & STABIL.?--NEG. MIX HT TO STOP
    1000.000          3
SOURCE STRNGTH (G/S)?
    100.0000
WIND SPD(M/S) & DIR. AT SITE            1 FOR HR            13
    3.000000       90.00000
WIND SPD(M/S) & DIR. AT SITE            2 FOR HR            13
    3.000000       90.00000
GRADIENT WIND SPD. (M/S) & DIR.                  FOR HR.            13?
    3.000000       90.00000
FOR HR FROM            13 TO            14
```

INTERACTIVE OUTPUTS FOR SAMPLE PROBLEM 1

HOUR NO. 0

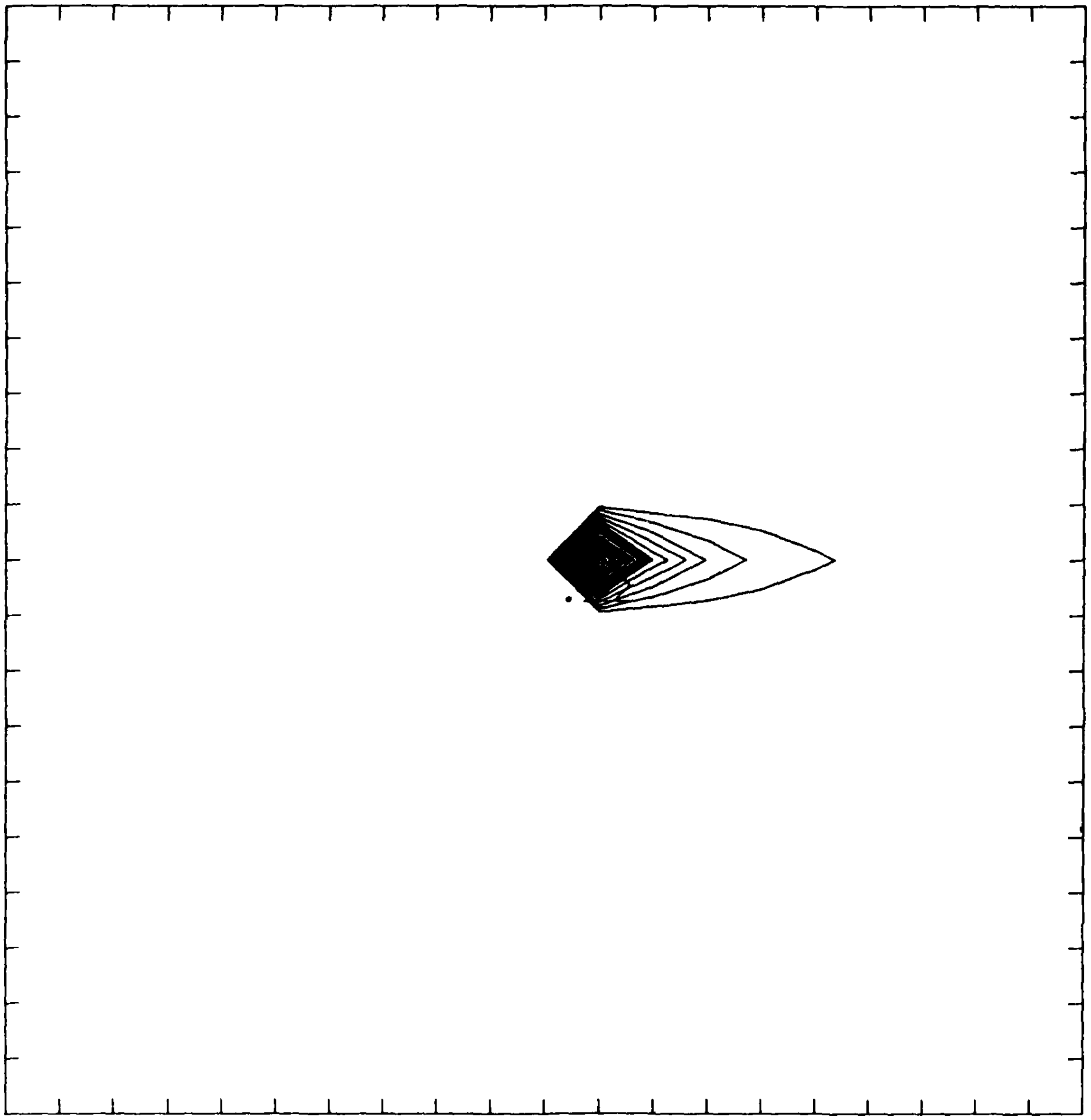
0.0000000E+00	0.0000000E+00	4.0989104E-05
300.0000	-300.0000	4.4592912E-06
300.0000	0.0000000E+00	0.1215399
300.0000	300.0000	4.4592912E-06
600.0000	-300.0000	6.5945397E-04
600.0000	0.0000000E+00	4.0102653E-02
600.0000	300.0000	6.5945397E-04
900.0000	-600.0000	1.4345395E-05
900.0000	-300.0000	2.3362250E-03
900.0000	0.0000000E+00	2.0100504E-02
900.0000	300.0000	2.3362250E-03
900.0000	600.0000	1.4345395E-05
1200.000	-900.0000	8.6097720E-07
1200.000	-600.0000	9.1205358E-05
1200.000	-300.0000	2.9783724E-03
1200.000	0.0000000E+00	1.1557194E-02
1200.000	300.0000	2.9783724E-03
1200.000	600.0000	9.1205358E-05
1200.000	900.0000	8.6097720E-07
1500.000	-900.0000	7.5220373E-06
1500.000	-600.0000	2.4965761E-04
1500.000	-300.0000	3.0180104E-03
1500.000	0.0000000E+00	7.7821030E-03
1500.000	300.0000	3.0180104E-03
1500.000	600.0000	2.4965761E-04
1500.000	900.0000	7.5220373E-06
1800.000	-1200.000	1.0715900E-06
1800.000	-900.0000	2.7701275E-05
1800.000	-600.0000	4.2142812E-04
1800.000	-300.0000	2.7560450E-03
1800.000	0.0000000E+00	5.6386401E-03
1800.000	300.0000	2.7560450E-03
1800.000	600.0000	4.2142812E-04
1800.000	900.0000	2.7701275E-05
1800.000	1200.000	1.0715900E-06
2100.000	-1200.000	4.2717256E-06
2100.000	-900.0000	6.4490610E-05
2100.000	-600.0000	5.5896986E-04
2100.000	-300.0000	2.4807358E-03
2100.000	0.0000000E+00	4.1019227E-03
2100.000	300.0000	2.4807358E-03
2100.000	600.0000	5.5896986E-04
2100.000	900.0000	6.4490610E-05
2100.000	1200.000	4.2717256E-06
2400.000	-1500.000	6.8043528E-07
2400.000	-1200.000	1.0673152E-05
2400.000	-900.0000	1.0524095E-04
2400.000	-600.0000	6.6844089E-04
2400.000	-300.0000	2.2540255E-03
2400.000	0.0000000E+00	3.3504537E-03
2400.000	300.0000	2.2540255E-03
2400.000	600.0000	6.6844089E-04
2400.000	900.0000	1.0524095E-04
2400.000	1200.000	1.0673152E-05
2400.000	1500.000	6.8043528E-07
2700.000	-1500.000	1.6746657E-06
2700.000	-1200.000	1.7932891E-05

2700.000	-900.0000	1.3834282E-04
2700.000	-600.0000	6.6199800E-04
2700.000	-300.0000	1.8354295E-03
2700.000	0.0000000E+00	2.5654752E-03
2700.000	300.0000	1.8354295E-03
2700.000	600.0000	6.6199800E-04
2700.000	900.0000	1.3834282E-04
2700.000	1200.000	1.7932891E-05
2700.000	1500.000	1.6746657E-06
3000.000	-1500.000	2.6516673E-06
3000.000	-1200.000	2.3911922E-05
3000.000	-900.0000	1.5422027E-04
3000.000	-600.0000	6.0906931E-04
3000.000	-300.0000	1.4423032E-03
3000.000	0.0000000E+00	1.9368917E-03
3000.000	300.0000	1.4423032E-03
3000.000	600.0000	6.0906931E-04
3000.000	900.0000	1.5422027E-04
3000.000	1200.000	2.3911922E-05
3000.000	1500.000	2.6516673E-06
HOUR NO.	1	
300.0000	-300.0000	1.3869529E-02
600.0000	-900.0000	2.6216809E-05
600.0000	-600.0000	1.0559696E-02
600.0000	-300.0000	2.8182262E-06
900.0000	-1200.000	8.4069790E-05
900.0000	-900.0000	7.9306541E-03
900.0000	-600.0000	4.5295895E-04
1200.000	-1500.000	1.1634143E-04
1200.000	-1200.000	4.5532910E-03
1200.000	-900.0000	2.6777729E-03
1500.000	-2100.000	2.1632263E-07
1500.000	-1800.000	1.0396002E-04
1500.000	-1500.000	2.8546054E-03
1500.000	-1200.000	4.2721303E-03
1500.000	-900.0000	7.2528914E-05
1800.000	-2400.000	2.5153582E-07
1800.000	-2100.000	5.7380716E-05
1800.000	-1800.000	1.6051946E-03
1800.000	-1500.000	4.8344992E-03
1800.000	-1200.000	5.7063770E-04
1800.000	-900.0000	5.4761114E-07
2100.000	-2400.000	2.7831709E-05
2100.000	-2100.000	8.1220100E-04
2100.000	-1800.000	3.9779954E-03
2100.000	-1500.000	1.9264598E-03
2100.000	-1200.000	4.6739275E-05
2400.000	-2700.000	8.3622808E-06
2400.000	-2400.000	3.0350240E-04
2400.000	-2100.000	2.6385817E-03
2400.000	-1800.000	3.1368693E-03
2400.000	-1500.000	4.0533504E-04
2400.000	-1200.000	3.0037263E-06
2700.000	-3000.000	1.9880299E-06
2700.000	-2700.000	9.6721051E-05
2700.000	-2400.000	1.1309104E-03
2700.000	-2100.000	3.2357811E-03
2700.000	-1800.000	1.4508724E-03
2700.000	-1500.000	7.6118937E-05
3000.000	-3000.000	1.8121415E-05

3000.000	-2700.000	4.0848547E-04
3000.000	-2400.000	2.3390125E-03
3000.000	-2100.000	2.7836331E-03
3000.000	-1800.000	5.6779763E-04
3000.000	-1500.000	1.5112245E-05
HOUR NO.	2	
300.0000	-300.0000	1.3769216E-02
600.0000	-900.0000	1.1306039E-04
600.0000	-600.0000	1.1427743E-02
900.0000	-1500.000	2.3146340E-06
900.0000	-1200.000	5.3723343E-04
900.0000	-900.0000	7.8289853E-03
900.0000	-600.0000	1.1306039E-04
1200.000	-1800.000	2.0476700E-05
1200.000	-1500.000	9.9088135E-04
1200.000	-1200.000	5.8960570E-03
1200.000	-900.0000	5.3723343E-04
1500.000	-2400.000	1.6687291E-06
1500.000	-2100.000	7.1838323E-05
1500.000	-1800.000	1.1804927E-03
1500.000	-1500.000	4.1370192E-03
1500.000	-1200.000	9.9088135E-04
1500.000	-900.0000	2.3146340E-06
1800.000	-3000.000	1.1811596E-07
1800.000	-2700.000	6.9976927E-06
1800.000	-2400.000	1.3474283E-04
1800.000	-2100.000	1.3319304E-03
1800.000	-1800.000	3.4382972E-03
1800.000	-1500.000	1.1804927E-03
1800.000	-1200.000	2.0476700E-05
2100.000	-3000.000	1.8314462E-05
2100.000	-2700.000	2.0981474E-04
2100.000	-2400.000	1.2717173E-03
2100.000	-2100.000	2.7568990E-03
2100.000	-1800.000	1.3319304E-03
2100.000	-1500.000	7.1838323E-05
2400.000	-3000.000	2.6925668E-04
2400.000	-2700.000	1.1810422E-03
2400.000	-2400.000	2.2239331E-03
2400.000	-2100.000	1.2717173E-03
2400.000	-1800.000	1.3474283E-04
2400.000	-1500.000	1.6687291E-06
2700.000	-3000.000	1.1672525E-03
2700.000	-2700.000	1.8472918E-03
2700.000	-2400.000	1.1810422E-03
2700.000	-2100.000	2.0981474E-04
2700.000	-1800.000	6.9976927E-06
3000.000	-3000.000	1.5004211E-03
3000.000	-2700.000	1.1672525E-03
3000.000	-2400.000	2.6925668E-04
3000.000	-2100.000	1.8314462E-05
3000.000	-1800.000	1.1811596E-07
HOUR NO.	3	
-2700.000	-3000.000	5.2660316E-06
-2400.000	-3000.000	6.7348831E-04
-2400.000	-2700.000	1.0261722E-04
-2400.000	-2400.000	4.9508103E-06
-2100.000	-3000.000	5.5164858E-03
-2100.000	-2700.000	4.0219431E-03
-2100.000	-2400.000	1.1767640E-03

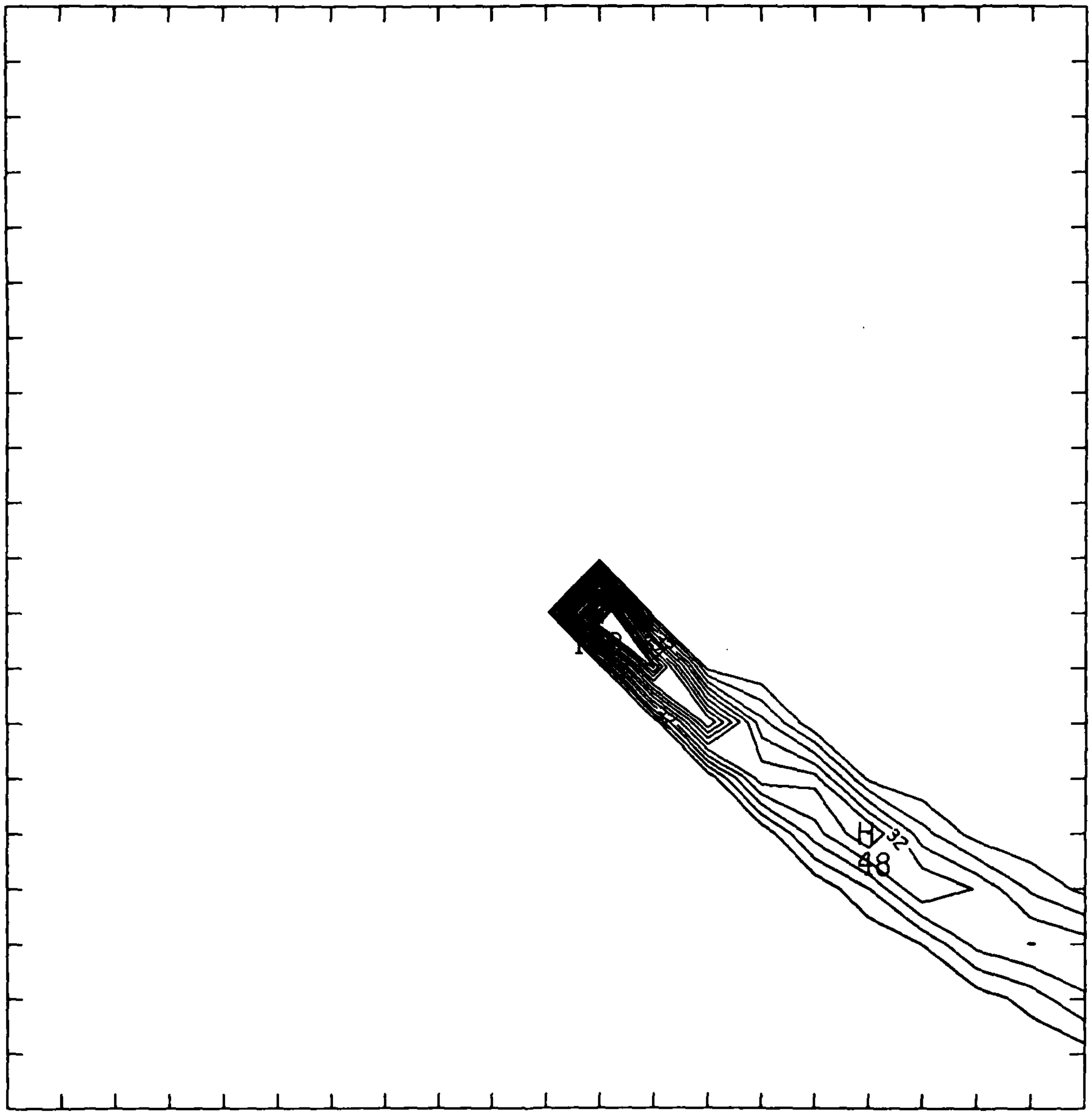
-2100.000	-2100.000	6.5625543E-05
-1800.000	-3000.000	1.8806105E-03
-1800.000	-2700.000	4.2788037E-03
-1800.000	-2400.000	6.6222851E-03
-1800.000	-2100.000	3.9531733E-03
-1800.000	-1800.000	2.5508928E-04
-1500.000	-3000.000	1.6144064E-05
-1500.000	-2700.000	7.4041724E-05
-1500.000	-2400.000	5.3887046E-04
-1500.000	-2100.000	4.1954191E-03
-1500.000	-1800.000	7.8165364E-03
-1500.000	-1500.000	1.0153673E-03
-1500.000	-1200.000	6.8185687E-07
-1200.000	-2100.000	2.1622223E-05
-1200.000	-1800.000	8.2986208E-04
-1200.000	-1500.000	8.2695642E-03
-1200.000	-1200.000	3.8602124E-03
-900.0000	-1500.000	1.0932298E-05
-900.0000	-1200.000	2.2898773E-03
-900.0000	-900.0000	5.1213251E-03
-600.0000	-900.0000	2.6873068E-04
-600.0000	-600.0000	6.4084125E-03
-300.0000	-300.0000	5.0510853E-03
HOUR NO.	4	
-3000.000	-3000.000	5.3017167E-03
-3000.000	-2700.000	1.0987022E-03
-3000.000	-2400.000	4.0536083E-06
-2700.000	-3000.000	1.0987022E-03
-2700.000	-2700.000	6.2017418E-03
-2700.000	-2400.000	8.1681402E-04
-2700.000	-2100.000	4.1324125E-07
-2400.000	-3000.000	4.0536083E-06
-2400.000	-2700.000	8.1681402E-04
-2400.000	-2400.000	6.5416615E-03
-2400.000	-2100.000	5.6641991E-04
-2100.000	-2700.000	4.1324125E-07
-2100.000	-2400.000	5.6641991E-04
-2100.000	-2100.000	7.0519042E-03
-2100.000	-1800.000	2.9203278E-04
-1800.000	-2100.000	2.9203278E-04
-1800.000	-1800.000	7.3660533E-03
-1800.000	-1500.000	1.0930931E-04
-1500.000	-1800.000	1.0930931E-04
-1500.000	-1500.000	7.9320269E-03
-1500.000	-1200.000	1.9031409E-05
-1200.000	-1500.000	1.9031409E-05
-1200.000	-1200.000	7.7571860E-03
-900.0000	-900.0000	6.9894125E-03
-600.0000	-600.0000	4.7408459E-03
HOUR NO.	5	
-3000.000	-1500.000	2.3354420E-07
-3000.000	-1200.000	3.8028197E-06
-3000.000	-900.0000	3.7929083E-05
-3000.000	-600.0000	1.6636004E-04
-3000.000	-300.0000	2.5618740E-04
-3000.000	0.0000000E+00	1.2025906E-04
-3000.000	300.0000	1.5757696E-05
-3000.000	600.0000	6.3537664E-07
-2700.000	-1500.000	1.5941145E-08
-2700.000	-1200.000	1.0907934E-06

-2700.000	-900.0000	1.9492374E-05
-2700.000	-600.0000	1.3639490E-04
-2700.000	-300.0000	3.0719966E-04
-2700.000	0.0000000E+00	1.6651506E-04
-2700.000	300.0000	1.8840337E-05
-2700.000	600.0000	5.1259138E-07
-2400.000	-1200.000	1.7081088E-07
-2400.000	-900.0000	7.0833871E-06
-2400.000	-600.0000	1.0008182E-04
-2400.000	-300.0000	3.6898456E-04
-2400.000	0.0000000E+00	2.3847843E-04
-2400.000	300.0000	2.3248505E-05
-2400.000	600.0000	3.8287607E-07
-2100.000	-900.0000	1.6643852E-06
-2100.000	-600.0000	5.5755696E-05
-2100.000	-300.0000	3.9303079E-04
-2100.000	0.0000000E+00	3.3935244E-04
-2100.000	300.0000	2.8948047E-05
-2100.000	600.0000	2.4422511E-07
-1800.000	-900.0000	1.6376549E-07
-1800.000	-600.0000	1.9398094E-05
-1800.000	-300.0000	3.6728318E-04
-1800.000	0.0000000E+00	5.1807280E-04
-1800.000	300.0000	3.2400396E-05
-1800.000	600.0000	8.3304762E-08
-1500.000	-600.0000	3.6264360E-06
-1500.000	-300.0000	2.9707851E-04
-1500.000	0.0000000E+00	6.6875864E-04
-1500.000	300.0000	2.4088988E-05
-1200.000	-600.0000	2.6448478E-07
-1200.000	-300.0000	1.9472046E-04
-1200.000	0.0000000E+00	9.4999908E-04
-1200.000	300.0000	1.0736760E-05
-900.0000	-300.0000	7.3852345E-05
-900.0000	0.0000000E+00	1.6132850E-03
-900.0000	300.0000	2.1092014E-06
-600.0000	-300.0000	3.2262963E-06
-600.0000	0.0000000E+00	2.6875243E-03
-300.0000	0.0000000E+00	4.9414118E-03



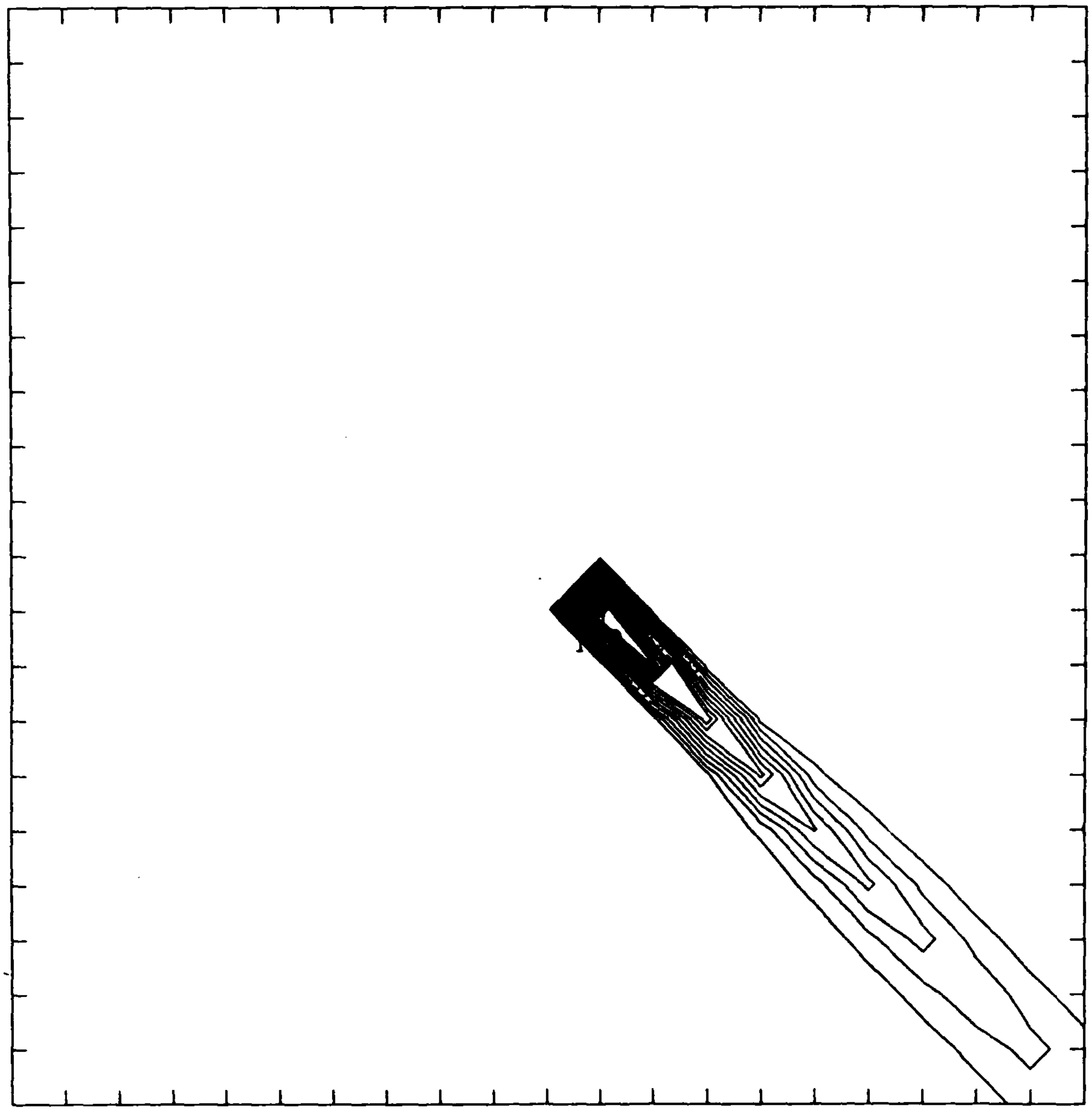
CØNTØUR FRØM 0.00000E+00 TØ 0.11900 CØNTØUR INTERVAL ØF 0.70000E-02 PT(3,3)= 0.00000E+00

FIGURE 4(a) GRAPHICAL OUTPUTS FOR SAMPLE PROBLEM 1



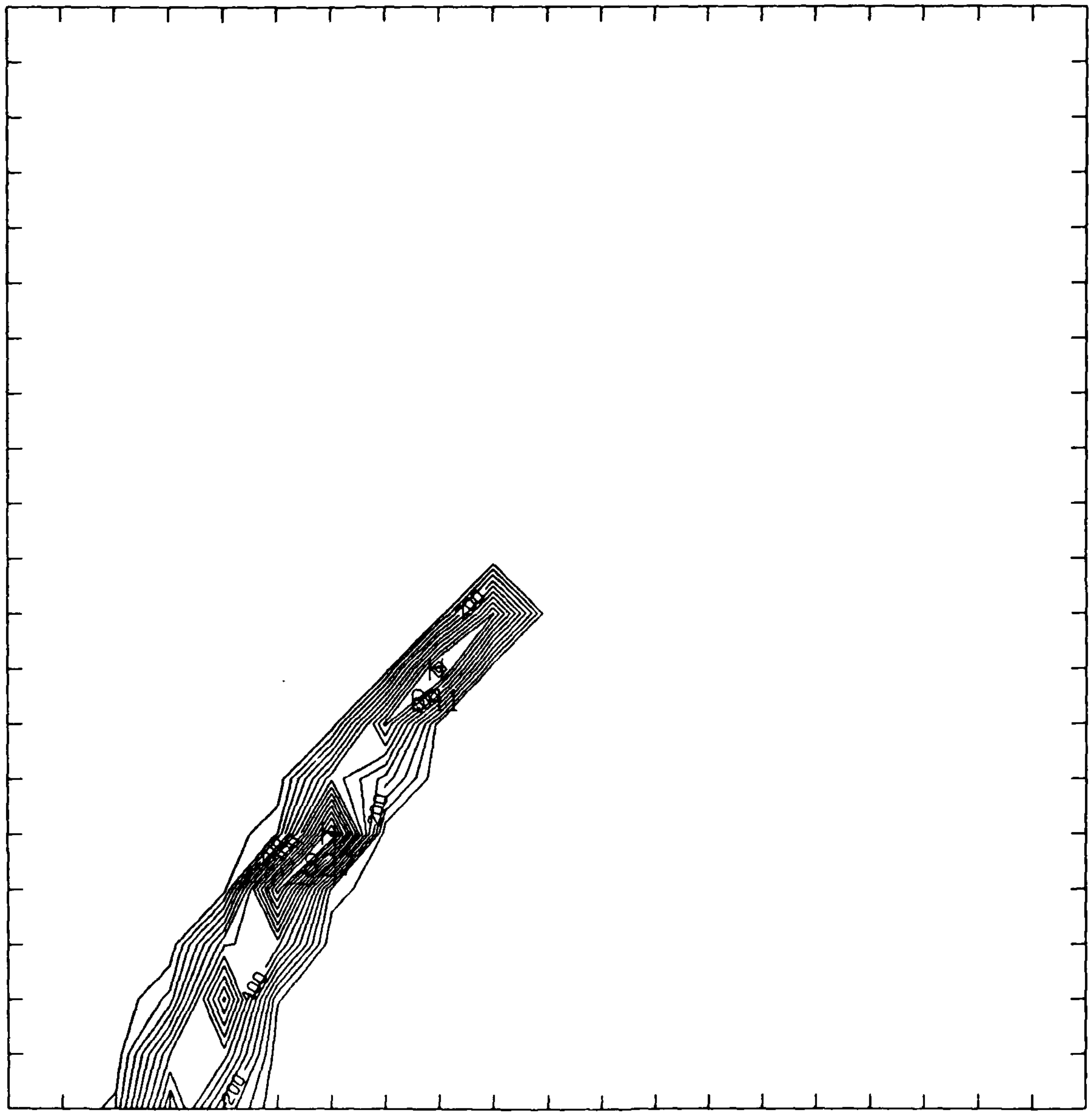
CØNTØUR FRØM 0.00000E+00 TØ 0.13600E-01 CØNTØUR INTERVAL ØF 0.80000E-03 PT(3,3)= 0.00000E+00 LABELS SCALED BY 10000.

FIGURE 4(b) GRAPHICAL OUTPUTS FOR SAMPLE PROBLEM 1 (continued)



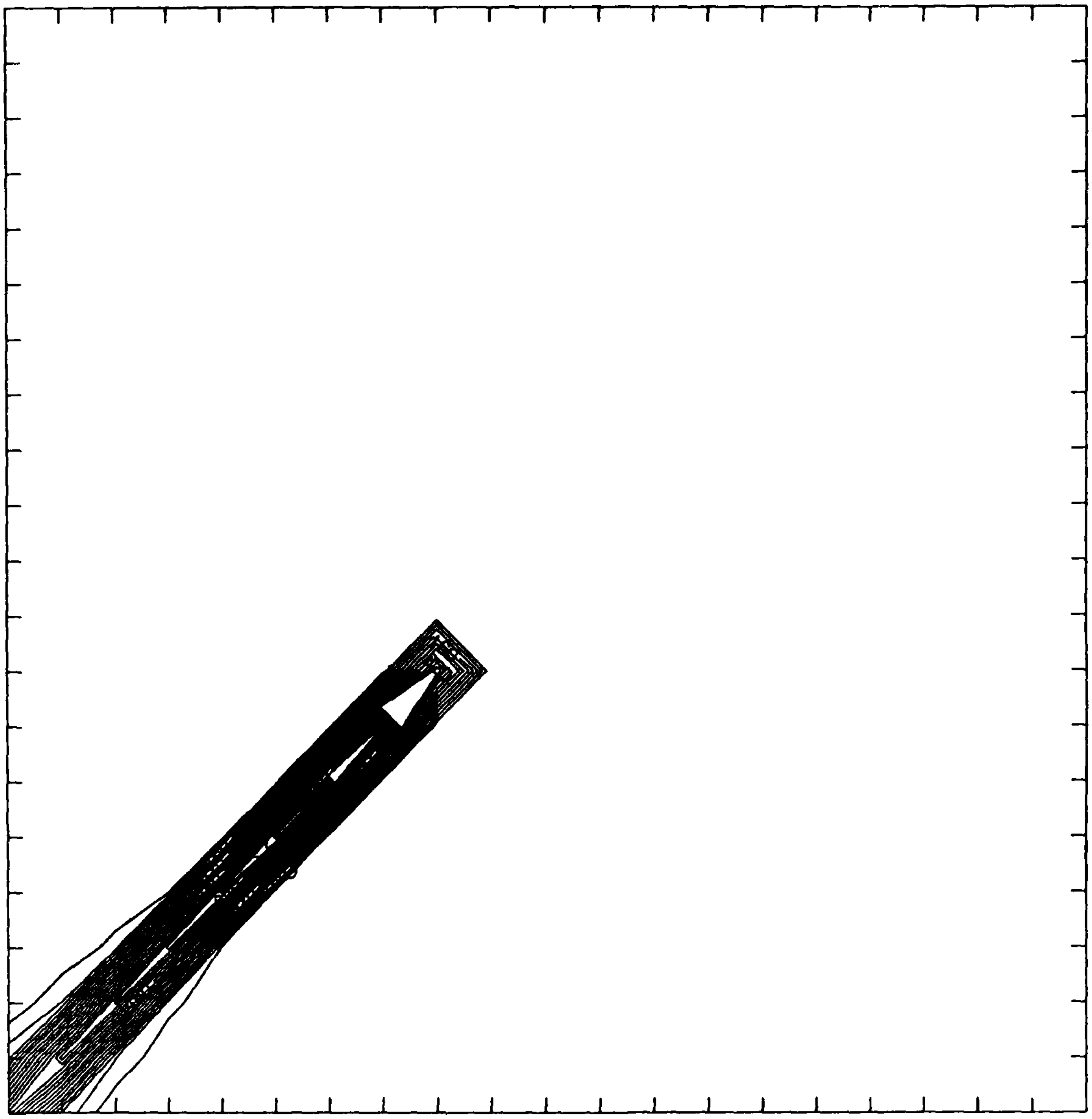
CØNTØUR FRØM 0.00000E+00 TØ 0.13600E-01 CØNTØUR INTERVAL ØF 0.80000E-03 PT(3,3)= 0.00000E+00 LABELS SCALED BY 10000.

FIGURE 4(c) GRAPHICAL OUTPUTS FOR SAMPLE PROBLEM 1 (continued)



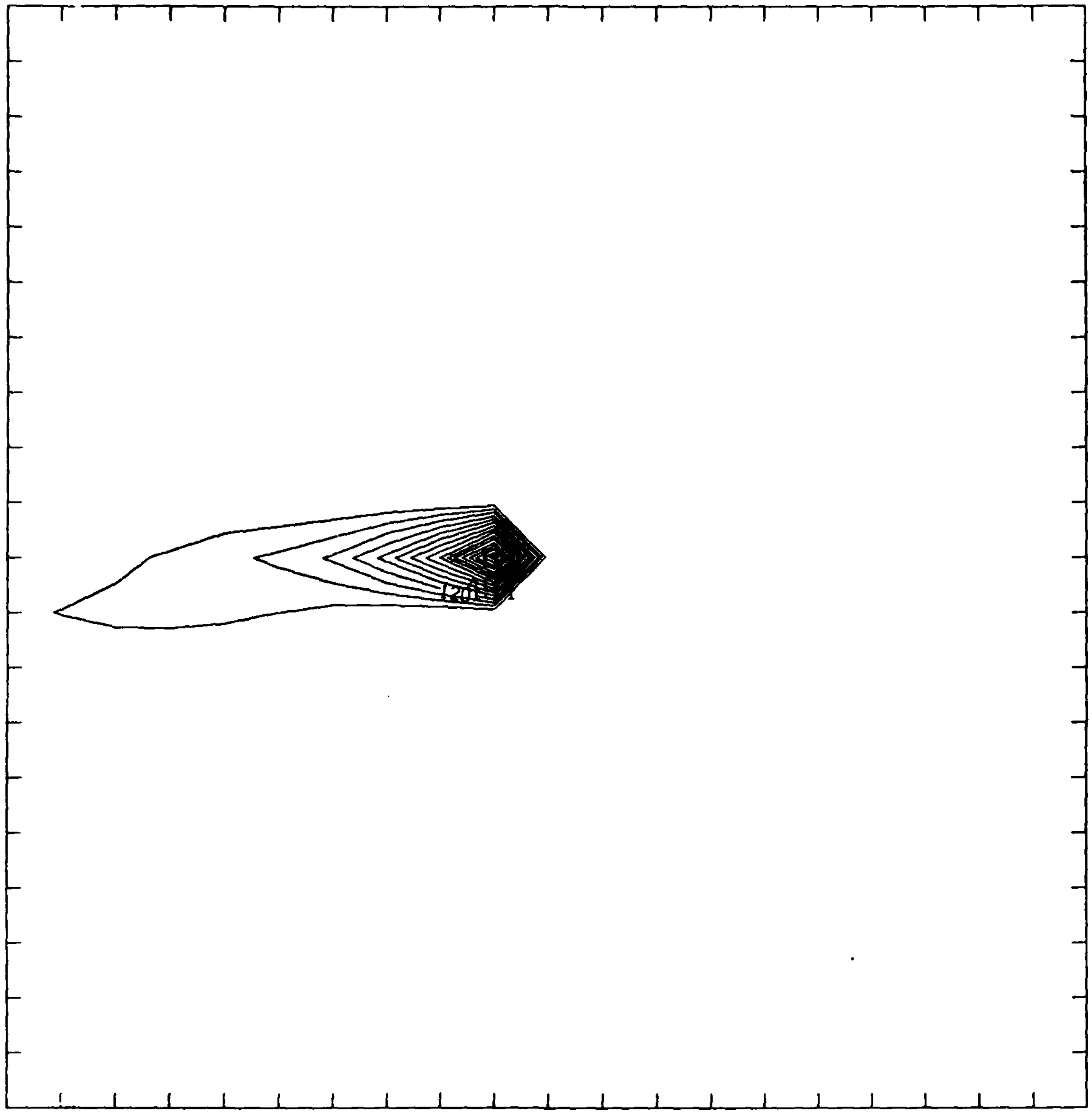
CØNTØUR FRØM 0.00000E+00 TØ 0.80000E-02 CØNTØUR INTERVAL ØF 0.50000E-03 PT(3,3)= 0.49508E-05 LABELS SCALED BY 0.10000E+06

FIGURE 4(d) GRAPHICAL OUTPUTS FOR SAMPLE PROBLEM 1 (continued)



CØNTOUR FRØM 0.00000E+00 TØ 0.76000E-02 CØNTØUR INTERVAL ØF 0.40000E-03 PT(3,3)= 0.65417E-02 LABELS SCALED BY 0.10000E+06

FIGURE 4(e) GRAPHICAL OUTPUTS FOR SAMPLE PROBLEM 1 (continued)



CØNTØUR FRØM 0.00000E+00 TØ 0.48000E-02 CØNTØUR INTERVAL ØF 0.30000E-03 PT(3,3)= 0.00000E+00 LABELS SCALED BY 0.10000E+06

FIGURE 4 (f) GRAPHICAL OUTPUTS FOR SAMPLE PROBLEM 1 (concluded)

10 0 2 60000. 0.6

431.87	-197.99	0.00	2191.24	455.09	-212.12	0.00	2125.89
450.80	-224.79	0.00	1916.68	441.34	-227.25	0.00	1846.45
449.31	-230.96	0.00	1961.06	456.91	-241.77	0.00	2082.49
472.17	-231.50	0.00	2222.94	472.64	-260.84	0.00	2462.88
461.54	-252.21	0.00	2820.35	456.59	-240.91	0.00	2927.15
423.02	-197.25	0.00	2367.78	415.40	-219.66	-5.30	2350.71
391.07	-240.43	-5.43	2083.95	398.76	-233.18	-6.34	2098.10
418.07	-240.56	-7.18	2239.52	425.36	-248.43	-4.98	2275.61
432.90	-246.00	-3.03	2524.82	472.77	-248.69	-13.18	2775.48
483.04	-282.44	-0.42	2623.33	465.69	-252.38	0.00	2912.03
444.29	-199.85	0.00	2496.53	425.42	-237.29	-1.79	2214.65
395.93	-246.46	1.18	2031.28	394.07	-246.80	1.35	2221.97
401.37	-237.37	-2.46	2379.98	414.85	-253.43	-4.16	2464.34
430.10	-255.79	2.69	2637.47	446.97	-234.52	-2.45	3072.97
457.77	-277.62	-13.38	2913.99	456.57	-267.34	0.00	2461.91
439.38	-195.80	0.00	2567.73	437.60	-235.11	-13.98	2188.81
409.21	-290.56	0.69	2033.24	391.49	-258.57	9.29	2229.28
401.08	-246.58	5.27	2481.90	407.66	-248.00	0.79	2600.41
428.61	-253.85	4.13	2759.39	451.92	-235.31	7.06	3031.52
471.29	-226.34	-11.74	2837.42	444.77	-296.35	0.00	2127.36
402.16	-235.79	0.00	2575.54	384.12	-209.36	-8.99	2577.00
426.74	-264.95	-16.35	2021.04	440.74	-304.92	7.59	2051.28
410.37	-257.95	17.51	2359.49	425.36	-236.40	9.33	2587.24
446.74	-237.96	4.46	2711.60	463.24	-221.20	8.49	2831.08
452.54	-217.17	-7.04	2891.55	428.30	-294.92	0.00	2373.15
432.49	-262.15	0.00	2227.82	401.98	-223.99	12.05	2295.12
375.66	-216.51	-5.52	2517.01	431.37	-261.71	-16.55	2063.47
489.06	-288.47	1.93	2012.75	450.90	-248.54	15.29	2402.90
453.13	-218.38	4.64	2712.09	458.47	-221.83	3.54	2785.24
464.23	-219.34	-6.31	2868.63	447.10	-272.47	0.00	2396.07
410.87	-254.29	0.00	2124.43	431.67	-248.09	4.62	2180.51
422.22	-231.15	13.66	2141.01	386.47	-223.48	3.23	2542.37
402.13	-262.71	-5.30	2455.57	450.40	-267.29	5.98	2313.16
454.29	-226.22	13.29	2731.59	447.37	-215.55	5.31	2773.05
465.86	-206.72	-5.42	2932.52	446.44	-262.10	0.00	2350.23
420.95	-249.32	0.00	2081.03	427.40	-227.76	2.84	2091.27
441.76	-239.63	1.16	2077.13	482.52	-227.61	8.39	2088.34
474.53	-218.14	6.25	2120.04	472.17	-248.67	3.89	2291.22
486.56	-235.94	12.44	2411.68	475.88	-215.72	9.08	2676.97
461.52	-202.12	3.88	2856.44	447.01	-245.83	0.00	2545.30
416.56	-226.58	0.00	1994.71	436.57	-222.41	3.04	2046.40
447.29	-218.22	2.95	2085.90	445.90	-215.32	2.43	2179.54
465.94	-226.81	-1.21	2278.54	487.86	-228.03	-0.42	2297.07
488.70	-242.57	3.49	2403.38	487.39	-230.06	8.26	2635.03
468.63	-214.73	8.78	2702.33	462.50	-227.75	0.00	2741.35
420.91	-201.68	0.00	1979.59	447.94	-202.21	0.00	1992.76
466.93	-204.34	0.00	2006.41	484.12	-211.12	0.00	2075.18
468.68	-206.86	0.00	2234.16	488.93	-225.47	0.00	2374.61
499.59	-234.75	0.00	2438.50	505.06	-231.20	0.00	2488.24
495.41	-227.33	0.00	2547.25	485.76	-233.19	0.00	2894.48
1010.39	-469.43	0.00	42729.25	1059.78	-488.55	0.00	41454.94
1074.77	-512.01	0.00	37375.25	1060.97	-511.23	0.00	36005.85
1049.38	-504.03	0.00	38240.64	1037.47	-512.94	0.00	40608.57
1040.29	-490.89	0.00	43347.39	1016.28	-536.19	0.00	48026.18
975.17	-515.07	0.00	54996.84	973.10	-497.21	0.00	57079.48
971.07	-449.11	0.00	46171.79	957.03	-484.54	-8.28	45838.94
939.31	-526.38	-8.61	40637.10	945.74	-506.12	-10.37	40912.89
957.50	-507.28	-11.50	43670.71	960.55	-515.85	-8.24	44374.44
947.83	-501.20	-5.48	49233.92	996.02	-498.89	-19.70	54121.94
1028.68	-560.58	-0.81	51154.90	987.51	-497.00	0.00	56784.67

993.77	-444.21	0.00	48682.36	983.62	-518.27	-2.75	43185.71
948.34	-537.95	1.91	39610.05	919.18	-523.44	1.73	43328.36
911.59	-495.55	-4.12	46409.53	923.38	-515.76	-6.74	48054.71
933.29	-510.94	2.85	51430.68	933.08	-460.48	-3.98	59922.89
964.11	-534.26	-18.70	56822.71	1000.08	-523.75	0.00	48007.17
977.31	-433.51	0.00	50070.79	1004.37	-515.45	-22.43	42681.70
966.51	-615.45	1.09	39648.09	909.75	-544.19	14.16	43471.01
899.24	-507.99	7.48	48397.06	897.50	-501.51	0.73	50707.94
919.64	-502.07	5.26	53808.12	941.59	-460.75	9.96	59114.57
990.59	-446.39	-15.72	55329.68	1008.02	-579.90	0.00	41483.47
910.05	-504.89	0.00	50222.94	872.49	-456.47	-13.81	50251.47
995.45	-574.58	-26.53	39410.34	1010.58	-636.92	11.97	39999.95
922.40	-534.94	26.27	46010.11	926.61	-484.39	13.38	50451.18
951.34	-479.20	6.12	52876.17	971.47	-443.62	11.98	55206.05
951.01	-430.06	-9.47	56385.27	950.51	-570.27	0.00	46276.39
993.49	-568.77	0.00	43442.48	926.35	-496.95	19.18	44754.82
854.81	-471.50	-8.43	49081.77	989.83	-566.41	-26.41	40237.70
1091.03	-609.82	2.79	39248.68	982.79	-516.89	22.39	46856.48
960.38	-449.91	6.27	52885.68	963.81	-449.31	4.91	54312.14
968.70	-439.24	-8.55	55938.30	976.46	-537.25	0.00	46723.34
965.97	-566.62	0.00	41426.41	987.78	-550.73	7.46	42520.04
968.29	-519.67	22.02	41749.74	864.26	-483.15	4.84	49576.27
893.82	-549.54	-7.95	47883.54	986.93	-557.95	8.95	45106.69
958.62	-467.39	18.98	53266.06	943.92	-444.29	7.49	54074.39
965.28	-421.83	-7.29	57184.08	976.65	-530.74	0.00	45829.43
987.11	-569.32	0.00	40580.05	987.68	-528.42	4.64	40779.75
1006.23	-545.81	1.88	40503.96	1069.30	-519.28	13.53	40722.69
1047.72	-495.64	9.91	41340.83	1023.20	-534.10	5.87	44678.75
1035.72	-502.87	18.58	47027.66	996.87	-454.41	12.97	52200.97
961.07	-423.17	5.72	55700.56	960.10	-505.20	0.00	49633.33
988.40	-549.66	0.00	38896.82	1005.85	-532.58	5.11	39904.85
1010.80	-518.15	4.89	40675.14	992.35	-501.91	3.95	42501.01
1013.14	-510.47	-1.88	44431.50	1046.55	-506.37	-0.67	44792.87
1037.36	-521.30	5.20	46866.00	1017.09	-487.17	12.10	51383.13
982.25	-456.28	12.84	52695.48	971.21	-475.84	0.00	53456.26
995.34	-519.19	0.00	38602.02	1027.16	-513.94	0.00	38858.78
1048.89	-510.12	0.00	39125.05	1064.26	-510.54	0.00	40465.92
1051.93	-487.49	0.00	43566.11	1037.11	-506.32	0.00	46304.92
1049.18	-514.01	0.00	47550.70	1055.69	-501.59	0.00	48520.69
1037.43	-489.92	0.00	49671.38	1000.47	-486.26	0.00	56442.32

10 10 2 60000. 0.6

-35.74	6.33	0.00	2191.24	-37.30	7.52	0.00	2125.89
-36.89	8.28	0.00	1916.68	-37.08	8.32	0.00	1846.45
-38.40	9.21	0.00	1961.06	-39.28	9.78	0.00	2082.49
-40.52	9.41	0.00	2222.94	-40.03	11.81	0.00	2462.88
-38.84	10.78	0.00	2820.35	-38.35	9.87	0.00	2927.15
-35.07	6.26	0.00	2367.78	-34.27	7.95	0.29	2350.71
-32.70	8.36	0.32	2083.95	-33.74	8.24	0.60	2098.10
-35.52	9.31	0.69	2239.52	-36.26	10.09	0.56	2275.61
-36.88	10.22	0.53	2524.82	-39.61	11.30	1.16	2775.48
-39.56	13.36	0.11	2623.33	-38.63	10.37	0.00	2912.03
-36.75	6.79	0.00	2496.53	-35.31	8.77	-0.10	2214.65
-33.24	8.58	-0.09	2031.28	-33.43	9.05	0.07	2221.97
-34.25	8.95	0.34	2379.98	-35.26	10.32	0.49	2464.34
-36.40	11.06	0.12	2637.47	-37.58	10.29	0.36	3072.97
-37.59	13.14	0.77	2913.99	-36.92	10.84	0.00	2461.91
-37.18	6.26	0.00	2567.73	-36.49	8.70	0.88	2188.81
-33.87	11.87	-0.04	2033.24	-33.00	9.83	-0.53	2229.28
-34.16	9.56	-0.23	2481.90	-34.72	10.05	0.09	2600.41
-36.12	11.00	-0.10	2759.39	-38.10	10.41	-0.54	3031.52
-39.42	9.58	0.47	2837.42	-36.44	12.78	0.00	2127.36

-33.88	8.96	0.00	2575.54	-32.93	6.65	0.45	2577.00
-35.56	10.22	1.08	2021.04	-36.29	13.51	-0.45	2051.28
-34.55	10.31	-1.16	2359.49	-36.15	9.34	-0.58	2587.24
-37.86	10.09	-0.23	2711.60	-39.10	9.30	-0.60	2631.08
-38.21	8.86	0.33	2891.55	-35.29	13.10	0.00	2373.15
-36.17	11.13	0.00	2227.82	-34.03	7.66	-0.85	2295.12
-32.14	6.96	0.34	2517.01	-35.83	10.14	1.11	2063.47
-40.20	12.97	0.01	2012.75	-37.72	10.36	-0.90	2402.90
-38.22	8.75	-0.17	2712.09	-38.73	9.22	-0.21	2785.24
-38.87	9.40	0.31	2868.63	-36.66	11.82	0.00	2396.07
-34.18	10.39	0.00	2124.43	-35.77	9.91	-0.38	2180.51
-35.13	8.17	-0.95	2141.01	-32.48	7.80	-0.10	2542.37
-32.91	10.76	0.32	2455.57	-36.82	11.60	-0.35	2313.16
-37.89	9.38	-0.86	2731.59	-37.44	8.59	-0.33	2773.05
-39.08	8.52	0.21	2932.52	-36.65	11.52	0.00	2350.23
-34.78	10.25	0.00	2081.03	-35.27	8.51	-0.24	2091.27
-36.05	9.18	-0.10	2077.13	-39.16	8.85	-0.67	2088.34
-38.92	7.94	-0.42	2120.04	-38.20	10.45	-0.17	2291.22
-39.58	10.08	-0.83	2411.68	-39.45	8.77	-0.52	2676.97
-38.50	7.99	-0.39	2856.44	-37.06	10.63	0.00	2545.30
-34.41	9.10	0.00	1994.71	-35.56	8.49	-0.21	2046.40
-36.06	8.13	-0.18	2085.90	-35.89	7.78	-0.11	2179.54
-37.19	8.91	0.16	2278.54	-39.03	9.33	0.10	2297.07
-39.20	10.35	-0.12	2403.38	-39.38	9.91	-0.53	2635.03
-38.77	8.36	-0.67	2702.33	-38.47	9.55	0.00	2741.35
-34.60	7.57	0.00	1979.59	-36.22	7.67	0.00	1992.76
-37.19	7.65	0.00	2006.41	-38.07	8.15	0.00	2075.18
-38.48	7.85	0.00	2234.16	-38.58	9.00	0.00	2374.61
-39.58	10.02	0.00	2438.50	-40.48	9.68	0.00	2488.24
-39.94	9.07	0.00	2547.25	-39.92	9.62	0.00	2894.48
-91.42	17.27	0.00	42729.25	-95.22	19.02	0.00	41454.94
-97.00	19.91	0.00	37375.25	-97.85	19.59	0.00	36005.85
-97.98	19.93	0.00	38240.64	-97.29	20.51	0.00	40608.57
-97.43	19.19	0.00	43347.39	-94.21	23.49	0.00	48026.18
-89.64	21.25	0.00	54996.84	-88.80	19.90	0.00	57079.48
-87.84	16.00	0.00	46171.79	-86.44	18.69	0.49	45838.94
-85.99	19.44	0.55	40637.10	-87.37	18.38	1.08	40912.89
-88.41	19.84	1.21	43670.71	-88.90	20.41	1.01	44374.44
-87.63	20.26	0.96	49233.92	-90.63	21.71	1.88	54121.94
-92.26	25.59	0.20	51154.90	-89.14	19.80	0.00	56784.67
-89.57	16.47	0.00	48682.36	-89.07	20.29	-0.18	43185.71
-86.97	19.68	-0.17	39610.05	-84.43	19.79	0.17	43328.36
-84.00	18.65	0.60	46409.53	-84.65	20.73	0.85	48054.71
-85.09	21.20	0.31	51430.68	-84.72	19.31	0.62	59922.89
-86.18	24.27	1.13	56822.71	-89.29	20.26	0.00	48007.17
-89.42	15.40	0.00	50070.79	-91.25	20.28	1.54	42681.70
-87.18	25.96	-0.06	39648.09	-83.12	21.16	-0.87	43471.01
-82.24	19.85	-0.34	48397.06	-81.97	19.83	0.19	50707.94
-83.38	20.96	-0.08	53808.12	-85.25	19.26	-0.83	59114.57
-89.91	17.60	0.59	55329.68	-91.40	23.37	0.00	41483.47
-83.25	20.58	0.00	50222.94	-80.68	16.10	0.74	50251.47
-90.39	23.34	1.92	39410.34	-90.27	28.65	-0.77	39999.95
-83.84	21.43	-1.89	46010.11	-84.53	18.83	-0.89	50451.18
-86.34	19.39	-0.33	52876.17	-88.42	17.39	-0.92	55206.05
-86.89	16.02	0.43	56385.27	-86.67	23.23	0.00	46276.39
-90.42	25.56	0.00	43442.48	-85.26	18.85	-1.47	44754.82
-78.94	16.69	0.55	49081.77	-89.76	22.88	1.93	40237.70
-97.43	27.32	0.06	39248.68	-88.88	21.10	-1.38	46856.48
-87.50	17.19	-0.21	52885.68	-87.74	17.32	-0.30	54312.14
-88.29	17.02	0.42	55938.30	-88.66	20.77	0.00	46723.34
-87.97	24.88	0.00	41426.41	-89.48	23.66	-0.66	42520.04

-88.38	20.13	-1.66	41749.74	-79.03	17.82	-0.15	49576.27
-80.41	22.74	0.51	47883.54	-88.49	23.58	-0.56	45106.69
-86.76	18.56	-1.30	53266.06	-86.18	16.40	-0.49	54074.39
-88.14	15.66	0.27	57184.08	-88.92	20.55	0.00	45829.43
-89.36	25.12	0.00	40580.05	-89.79	21.71	-0.42	40779.75
-90.78	22.40	-0.17	40503.96	-96.24	21.23	-1.19	40722.69
-94.94	18.42	-0.71	41340.83	-91.66	22.13	-0.26	44678.75
-93.07	20.42	-1.33	47027.66	-90.43	17.31	-0.77	52200.97
-87.91	15.17	-0.64	55700.56	-87.91	19.46	0.00	49633.33
-89.74	23.60	0.00	38896.82	-90.73	22.01	-0.38	39904.85
-91.01	20.85	-0.32	40675.14	-89.32	19.07	-0.18	42501.01
-90.58	20.45	0.28	44431.50	-93.51	20.20	0.18	44792.87
-92.55	21.55	-0.16	46866.00	-91.08	19.81	-0.82	51383.13
-89.45	16.56	-1.07	52695.48	-88.78	18.22	0.00	53456.26
-90.01	21.23	0.00	38602.02	-92.43	21.01	0.00	38858.78
-93.79	20.23	0.00	39125.05	-94.49	20.53	0.00	40465.92
-93.28	18.64	0.00	43566.11	-91.92	20.21	0.00	46304.92
-93.04	21.12	0.00	47550.70	-94.15	20.29	0.00	48520.69
-92.81	18.73	0.00	49671.38	-90.17	19.10	0.00	56442.32
10 9	2 60000.	0.6					
6.32	-28.95	0.00	2191.24	8.27	-28.26	0.00	2125.89
8.92	-28.64	0.00	1916.68	6.47	-27.92	0.00	1846.45
4.42	-26.90	0.00	1961.06	3.22	-27.82	0.00	2082.49
2.32	-27.77	0.00	2222.94	2.35	-28.80	0.00	2462.88
2.00	-31.13	0.00	2820.35	2.24	-31.45	0.00	2927.15
5.92	-28.34	0.00	2367.78	6.54	-29.07	-0.46	2350.71
6.16	-31.96	-0.41	2083.95	5.20	-30.35	0.10	2098.10
4.94	-28.58	0.13	2239.52	4.66	-29.18	0.30	2275.61
3.68	-28.88	0.65	2524.82	4.44	-27.60	0.03	2775.48
6.31	-30.09	0.19	2623.33	4.11	-31.51	0.00	2912.03
6.13	-27.23	0.00	2496.53	6.73	-30.44	-0.66	2214.65
5.96	-32.75	0.03	2031.28	5.40	-30.97	0.47	2221.97
4.79	-29.27	0.29	2379.98	5.18	-29.38	0.29	2464.34
5.31	-29.12	0.86	2637.47	4.34	-26.90	0.40	3072.97
6.24	-29.45	-1.02	2913.99	7.06	-33.08	0.00	2461.91
4.06	-27.59	0.00	2567.73	6.34	-29.79	-0.93	2188.81
8.27	-33.81	0.06	2033.24	6.05	-31.68	0.72	2229.28
5.22	-29.60	0.57	2481.90	5.24	-29.29	0.39	2600.41
5.64	-28.70	0.64	2759.39	4.89	-26.86	0.18	3031.52
4.51	-27.38	-1.42	2837.42	6.22	-34.72	0.00	2127.36
4.38	-29.56	0.00	2575.54	3.23	-29.87	-0.92	2577.00
7.36	-31.98	-0.94	2021.04	9.63	-32.26	0.55	2051.28
6.27	-30.38	0.95	2359.49	5.10	-28.57	0.59	2587.24
5.51	-27.82	0.39	2711.60	4.71	-26.87	0.34	2831.08
3.78	-27.68	-0.73	2891.55	5.15	-34.64	0.00	2373.15
6.19	-29.17	0.00	2227.82	4.58	-29.93	0.58	2295.12
3.74	-30.64	-0.40	2517.01	7.39	-31.50	-0.91	2063.47
9.79	-30.27	0.44	2012.75	6.59	-28.99	1.13	2402.90
4.83	-27.29	0.58	2712.09	4.83	-27.70	0.26	2785.24
4.41	-27.30	-0.62	2868.63	5.91	-33.91	0.00	2396.07
6.79	-29.52	0.00	2124.43	7.09	-29.62	0.10	2180.51
6.18	-30.30	0.70	2141.01	4.74	-30.50	0.49	2542.37
6.92	-31.51	-0.38	2455.57	7.85	-30.83	0.45	2313.16
5.53	-27.88	0.80	2731.59	4.49	-28.43	0.34	2773.05
4.12	-27.18	-0.67	2932.52	5.99	-33.09	0.00	2350.23
7.89	-29.60	0.00	2081.03	7.05	-29.28	0.05	2091.27
7.83	-30.47	0.02	2077.13	7.97	-28.37	0.20	2088.34
6.72	-29.46	0.36	2120.04	7.76	-29.96	0.44	2291.22
7.20	-28.87	0.69	2411.68	5.38	-28.19	0.73	2676.97
4.08	-27.84	-0.14	2856.44	4.81	-31.87	0.00	2545.30
8.13	-29.19	0.00	1994.71	8.22	-29.60	0.18	2046.40

7.97	-29.52	0.23	2085.90	7.37	-30.18	0.29	2179.54
7.82	-29.61	0.14	2278.54	7.96	-29.35	0.16	2297.07
7.97	-29.86	0.50	2403.38	7.19	-28.71	0.52	2635.03
4.95	-29.80	0.26	2702.33	4.54	-30.19	0.00	2741.35
8.83	-29.32	0.00	1979.59	8.71	-28.94	0.00	1992.76
9.03	-29.32	0.00	2006.41	9.50	-29.16	0.00	2075.18
8.96	-29.24	0.00	2234.16	8.57	-29.97	0.00	2374.61
8.60	-29.71	0.00	2438.50	8.01	-29.36	0.00	2488.24
7.84	-30.45	0.00	2547.25	6.24	-30.62	0.00	2894.48
-25.06	-74.85	0.00	42729.25	-23.95	-74.27	0.00	41454.94
-25.77	-77.85	0.00	37375.25	-32.62	-78.70	0.00	36005.85
-36.42	-75.48	0.00	38240.64	-37.52	-76.08	0.00	40608.57
-37.26	-74.37	0.00	43347.39	-33.15	-72.63	0.00	48026.18
-29.22	-74.35	0.00	54996.84	-26.90	-73.94	0.00	57079.48
-24.03	-73.35	0.00	46171.79	-23.21	-74.27	-0.69	45838.94
-26.70	-83.59	-0.58	40637.10	-29.60	-81.44	0.76	40912.89
-29.83	-76.76	0.86	43670.71	-30.66	-77.25	1.12	44374.44
-29.98	-73.61	1.73	49233.92	-26.57	-68.04	0.82	54121.94
-22.86	-72.38	0.46	51154.90	-23.96	-74.09	0.00	56784.67
-23.47	-70.34	0.00	48682.36	-24.66	-78.38	-1.41	43185.71
-27.50	-86.09	-0.03	39610.05	-27.13	-81.55	1.06	43328.36
-27.80	-77.51	0.88	46409.53	-26.69	-76.14	0.99	48054.71
-25.38	-73.16	1.86	51430.68	-24.25	-65.25	0.98	59922.89
-20.43	-69.34	-1.46	56822.71	-20.87	-80.08	0.00	48007.17
-27.67	-70.96	0.00	50070.79	-26.20	-77.61	-1.12	42681.70
-23.01	-86.70	0.10	39648.09	-25.32	-82.81	1.06	43471.01
-25.28	-76.95	0.99	48397.06	-24.87	-75.18	0.86	50707.94
-23.40	-71.43	1.23	53808.12	-23.60	-65.48	-0.01	59114.57
-25.65	-67.08	-2.47	55329.68	-25.85	-85.40	0.00	41483.47
-25.68	-73.61	0.00	50222.94	-27.62	-75.18	-1.41	50251.47
-25.79	-84.04	-0.98	39410.34	-21.04	-83.62	0.76	39999.95
-24.16	-79.25	1.02	46010.11	-25.34	-73.75	0.77	50451.18
-24.50	-70.32	0.62	52876.17	-25.90	-67.02	0.26	55206.05
-26.60	-67.84	-1.21	56385.27	-26.04	-83.17	0.00	46276.39
-26.41	-74.78	0.00	43442.48	-27.93	-77.32	0.42	44754.82
-26.66	-77.47	-0.52	49081.77	-25.21	-83.32	-0.92	40237.70
-22.61	-80.48	1.00	39248.68	-24.19	-75.24	1.64	46856.48
-25.89	-69.53	1.05	52885.68	-25.76	-68.90	0.37	54312.14
-26.33	-67.06	-1.02	55938.30	-25.44	-82.22	0.00	46723.34
-25.83	-76.57	0.00	41426.41	-24.83	-76.23	-0.14	42520.04
-26.61	-79.46	0.58	41749.74	-24.29	-76.98	0.87	49576.27
-20.55	-79.02	-0.52	47883.54	-21.68	-78.51	0.65	45106.69
-23.85	-69.86	0.97	53266.06	-26.08	-70.68	0.45	54074.39
-26.76	-66.80	-1.16	57184.08	-26.27	-81.17	0.00	45829.43
-24.51	-76.58	0.00	40580.05	-25.89	-76.59	-0.13	40779.75
-24.02	-79.22	-0.06	40503.96	-24.76	-75.63	-0.21	40722.69
-26.47	-78.52	0.39	41340.83	-22.19	-76.75	0.77	44678.75
-23.46	-73.65	0.75	47027.66	-25.51	-70.49	1.08	52200.97
-27.14	-68.90	-0.57	55700.56	-27.48	-77.64	0.00	49633.33
-25.42	-76.24	0.00	38896.82	-23.89	-76.84	0.16	39904.85
-23.51	-76.69	0.30	40675.14	-22.99	-78.16	0.51	42501.01
-21.39	-75.83	0.42	44431.50	-21.89	-75.52	0.41	44792.87
-21.20	-75.11	0.94	46866.00	-21.95	-71.30	0.64	51383.13
-26.35	-73.64	0.01	52695.48	-27.26	-73.55	0.00	53456.26
-24.34	-75.82	0.00	38602.02	-23.66	-75.03	0.00	38858.78
-22.16	-76.60	0.00	39125.05	-20.18	-75.66	0.00	40465.92
-19.58	-75.54	0.00	43566.11	-19.15	-75.50	0.00	46304.92
-19.52	-74.55	0.00	47550.70	-21.48	-73.84	0.00	48520.69
-22.11	-75.87	0.00	49671.38	-23.63	-73.55	0.00	56442.32
10 8	2 60000.	0.6					
27.96	-9.84	0.00	2191.24	29.20	-10.50	0.00	2125.89

31.01	-11.77	0.00	1916.68	32.58	-12.22	0.00	1846.45
32.95	-11.93	0.00	1961.06	32.76	-11.19	0.00	2082.49
32.11	-10.34	0.00	2222.94	29.73	-9.85	0.00	2462.88
26.63	-7.65	0.00	2820.35	25.05	-6.74	0.00	2927.15
26.38	-8.87	0.00	2367.78	26.52	-9.18	-0.26	2350.71
28.14	-9.97	-0.31	2083.95	29.25	-10.01	-0.49	2098.10
29.70	-9.81	-0.55	2239.52	30.12	-9.73	-0.45	2275.61
28.97	-8.61	-0.40	2524.82	27.80	-8.25	-0.83	2775.48
27.04	-8.79	-0.07	2623.33	24.88	-6.20	0.00	2912.03
25.94	-8.53	0.00	2496.53	27.40	-9.73	0.00	2214.65
28.53	-10.00	0.08	2031.28	28.22	-9.22	-0.02	2221.97
28.37	-8.65	-0.25	2379.98	28.65	-8.52	-0.38	2464.34
28.12	-7.99	-0.10	2637.47	26.08	-6.55	-0.27	3072.97
25.04	-7.24	-0.48	2913.99	25.35	-6.95	0.00	2461.91
25.62	-7.79	0.00	2567.73	27.67	-9.61	-0.76	2188.81
28.13	-10.88	0.04	2033.24	27.42	-9.25	0.47	2229.28
27.34	-7.95	0.19	2481.90	27.58	-7.59	-0.07	2600.41
27.23	-7.21	0.05	2759.39	26.26	-6.22	0.36	3031.52
26.34	-6.11	-0.25	2837.42	27.12	-8.16	0.00	2127.36
23.79	-8.21	0.00	2575.54	24.21	-6.85	-0.35	2577.00
28.29	-10.35	-0.97	2021.04	29.02	-11.24	0.42	2051.28
27.50	-8.52	0.92	2359.49	27.49	-7.10	0.42	2587.24
27.79	-6.85	0.16	2711.60	27.26	-6.07	0.40	2831.08
25.57	-5.19	-0.17	2891.55	25.14	-6.82	0.00	2373.15
25.40	-10.42	0.00	2227.82	25.18	-8.15	0.64	2295.12
24.24	-6.70	-0.25	2517.01	28.05	-9.82	-0.96	2063.47
30.57	-11.03	0.04	2012.75	28.38	-8.01	0.69	2402.90
27.16	-6.25	0.10	2712.09	26.67	-5.74	0.13	2785.24
25.44	-5.39	-0.16	2868.63	25.13	-6.12	0.00	2396.07
24.47	-10.39	0.00	2124.43	25.25	-9.29	0.27	2180.51
25.97	-8.68	0.75	2141.01	23.89	-6.73	0.08	2542.37
24.46	-7.73	-0.24	2455.57	26.87	-8.43	0.28	2313.16
26.16	-6.19	0.56	2731.59	25.67	-5.32	0.21	2773.05
24.91	-4.79	-0.09	2932.52	25.21	-6.30	0.00	2350.23
23.96	-10.31	0.00	2081.03	24.74	-9.10	0.17	2091.27
25.61	-9.09	0.07	2077.13	27.21	-9.33	0.50	2088.34
27.91	-8.53	0.34	2120.04	26.93	-8.11	0.14	2291.22
26.85	-7.36	0.60	2411.68	25.89	-5.70	0.32	2676.97
24.60	-4.66	0.26	2856.44	24.40	-5.49	0.00	2545.30
23.51	-10.16	0.00	1994.71	24.16	-9.02	0.16	2046.40
24.64	-8.60	0.15	2085.90	24.75	-7.94	0.10	2179.54
25.09	-7.79	-0.10	2278.54	26.07	-7.80	-0.06	2297.07
25.96	-7.24	0.09	2403.38	24.96	-6.12	0.34	2635.03
24.54	-4.85	0.42	2702.33	23.80	-4.75	0.00	2741.35
22.92	-9.10	0.00	1979.59	24.16	-8.84	0.00	1992.76
24.87	-8.69	0.00	2006.41	25.19	-8.43	0.00	2075.18
25.06	-7.55	0.00	2234.16	24.86	-6.97	0.00	2374.61
25.09	-6.95	0.00	2438.50	25.29	-6.49	0.00	2488.24
24.55	-5.48	0.00	2547.25	22.87	-4.20	0.00	2894.48
-1.06	37.94	0.00	42729.25	-2.09	37.93	0.00	41454.94
-1.35	39.73	0.00	37375.25	1.90	40.08	0.00	36005.85
3.72	38.61	0.00	38240.64	4.34	39.02	0.00	40608.57
4.13	38.02	0.00	43347.39	2.49	37.88	0.00	48026.18
1.10	38.43	0.00	54996.84	-0.02	38.05	0.00	57079.48
-1.08	37.07	0.00	46171.79	-1.21	37.93	0.40	45838.94
0.66	42.47	0.35	40637.10	1.90	41.30	-0.21	40912.89
1.88	39.32	-0.24	43670.71	2.23	39.65	-0.39	44374.44
2.01	37.92	-0.69	49233.92	-0.21	35.50	-0.13	54121.94
-2.31	38.14	-0.19	51154.90	-1.48	38.14	0.00	56784.67
-1.66	35.72	0.00	48682.36	-0.90	40.12	0.64	43185.71
0.89	43.70	-0.01	39610.05	1.09	41.61	-0.48	43328.36

1.50	39.55	-0.33	46409.53	0.89	39.23	-0.35	48054.71
0.15	37.90	-0.84	51430.68	-0.50	33.88	-0.38	59922.89
-2.63	36.57	0.86	56822.71	-2.90	41.05	0.00	48007.17
0.33	35.87	0.00	50070.79	-0.51	39.75	0.75	42681.70
-1.32	44.95	-0.06	39648.09	0.39	42.43	-0.63	43471.01
0.50	39.50	-0.52	48397.06	0.36	38.69	-0.38	50707.94
-0.59	37.09	-0.60	53808.12	-0.89	34.02	-0.11	59114.57
-0.68	34.52	1.26	55329.68	-0.74	44.06	0.00	41483.47
0.27	37.93	0.00	50222.94	1.63	38.04	0.78	50251.47
-0.50	43.28	0.74	39410.34	-2.71	43.92	-0.47	39999.95
-0.28	40.83	-0.75	46010.11	0.17	37.87	-0.49	50451.18
-0.52	36.35	-0.34	52876.17	-0.23	34.49	-0.26	55206.05
0.22	34.69	0.64	56385.27	-0.01	43.07	0.00	46276.39
-0.42	39.17	0.00	43442.48	1.12	39.43	-0.41	44754.82
1.47	39.25	0.33	49081.77	-0.67	42.92	0.71	40237.70
-2.99	42.25	-0.47	39248.68	-1.02	38.91	-0.98	46856.48
-0.06	35.66	-0.53	52885.68	-0.21	35.41	-0.22	54312.14
-0.12	34.50	0.55	55938.30	-0.61	42.29	0.00	46723.34
-0.34	39.92	0.00	41426.41	-1.01	39.62	-0.03	42520.04
0.09	40.66	-0.51	41749.74	0.33	39.22	-0.44	49576.27
-1.60	40.96	0.32	47883.54	-2.19	40.85	-0.39	45106.69
-0.97	36.04	-0.65	53266.06	0.15	36.14	-0.28	54074.39
0.09	34.19	0.59	57184.08	-0.24	41.77	0.00	45829.43
-1.21	39.98	0.00	40580.05	-0.55	39.51	0.00	40779.75
-1.54	40.89	0.00	40503.96	-1.95	39.02	-0.07	40722.69
-0.89	40.02	-0.29	41340.83	-2.46	39.79	-0.41	44678.75
-2.07	38.09	-0.55	47027.66	-0.77	36.17	-0.63	52200.97
0.30	35.12	0.18	55700.56	0.45	39.95	0.00	49633.33
-0.85	39.61	0.00	38896.82	-1.67	39.71	-0.13	39904.85
-1.84	39.49	-0.19	40675.14	-1.81	39.95	-0.27	42501.01
-2.75	39.08	-0.16	44431.50	-2.92	38.91	-0.17	44792.87
-3.12	38.97	-0.47	46866.00	-2.60	36.93	-0.42	51383.13
-0.30	37.59	-0.16	52695.48	0.16	37.83	0.00	53456.26
-1.44	39.12	0.00	38602.02	-2.04	38.73	0.00	38858.78
-2.90	39.37	0.00	39125.05	-3.92	38.98	0.00	40465.92
-4.03	38.68	0.00	43566.11	-4.04	38.93	0.00	46304.92
-4.03	38.64	0.00	47550.70	-3.29	38.20	0.00	48520.69
-2.84	38.96	0.00	49671.38	-1.86	37.98	0.00	56442.32

10 7 2 60000. 0.6

-2.66	8.77	0.00	2191.24	-3.64	8.36	0.00	2125.89
-4.26	8.36	0.00	1916.68	-4.41	8.06	0.00	1846.45
-4.99	7.89	0.00	1961.06	-5.79	8.23	0.00	2082.49
-7.09	8.30	0.00	2222.94	-8.68	9.25	0.00	2462.88
-9.91	9.79	0.00	2820.35	-10.89	9.79	0.00	2927.15
-2.07	8.62	0.00	2367.78	-2.73	8.38	0.14	2350.71
-2.95	8.85	0.13	2083.95	-3.63	8.35	0.03	2098.10
-4.67	8.05	0.05	2239.52	-5.59	8.37	0.02	2275.61
-6.64	8.34	-0.04	2524.82	-8.93	8.22	0.25	2775.48
-10.95	9.07	-0.01	2623.33	-11.19	8.70	0.00	2912.03
-1.65	8.25	0.00	2496.53	-2.14	8.60	0.17	2214.65
-2.38	8.79	-0.01	2031.28	-3.32	8.29	-0.11	2221.97
-4.19	7.93	-0.03	2379.98	-5.22	8.18	0.00	2464.34
-6.46	8.22	-0.16	2637.47	-7.81	7.38	-0.01	3072.97
-9.81	8.14	0.39	2913.99	-11.15	7.82	0.00	2461.91
-0.44	8.14	0.00	2567.73	-1.35	8.20	0.24	2183.81
-2.39	9.05	-0.02	2033.24	-2.90	8.29	-0.20	2229.28
-3.82	7.79	-0.15	2481.90	-4.70	7.85	-0.07	2600.41
-5.77	7.78	-0.15	2759.39	-7.07	6.99	-0.12	3031.52
-8.72	6.29	0.37	2837.42	-10.02	7.17	0.00	2127.36
0.50	8.61	0.00	2575.54	-0.06	7.89	0.23	2577.00
-1.53	8.21	0.23	2021.04	-3.21	8.39	-0.14	2051.28

-3.45	7.76	-0.29	2359.49	-4.13	7.27	-0.18	2587.24
-5.11	7.09	-0.10	2711.60	-5.96	6.38	-0.14	2831.08
-6.90	5.78	0.19	2691.55	-8.12	6.71	0.00	2373.15
1.33	8.56	0.00	2227.82	0.76	7.76	-0.11	2295.12
-0.23	7.48	0.10	2517.01	-1.86	7.65	0.22	2063.47
-3.67	7.52	-0.10	2012.75	-3.79	7.07	-0.30	2402.90
-4.02	6.43	-0.13	2712.09	-4.64	6.19	-0.07	2785.24
-5.67	5.51	0.15	2868.63	-7.00	5.87	0.00	2396.07
2.50	8.43	0.00	2124.43	1.18	7.57	0.01	2180.51
0.36	7.13	-0.12	2141.01	-0.68	7.08	-0.11	2542.37
-2.01	7.43	0.09	2455.57	-3.01	7.22	-0.11	2313.16
-3.20	6.30	-0.21	2731.59	-3.26	6.04	-0.09	2773.05
-4.23	5.23	0.15	2932.52	-5.44	5.44	0.00	2350.23
3.41	8.15	0.00	2081.03	2.34	7.07	0.02	2091.27
0.91	6.93	0.00	2077.13	-0.46	6.22	-0.01	2088.34
-1.16	6.35	-0.07	2120.04	-2.20	6.60	-0.10	2291.22
-2.58	6.20	-0.16	2411.68	-2.57	5.78	-0.16	2676.97
-2.91	5.20	0.01	2856.44	-3.85	5.33	0.00	2545.30
4.71	7.33	0.00	1994.71	3.00	6.67	-0.01	2046.40
1.56	6.27	-0.03	2085.90	0.38	6.36	-0.06	2179.54
-0.80	6.27	-0.04	2278.54	-1.56	6.14	-0.04	2297.07
-1.96	6.21	-0.11	2403.38	-2.16	5.83	-0.11	2635.03
-2.08	5.68	-0.06	2702.33	-2.77	5.18	0.00	2741.35
5.50	6.15	0.00	1979.59	3.46	5.61	0.00	1992.76
1.76	5.77	0.00	2006.41	0.36	5.84	0.00	2075.18
-0.65	5.91	0.00	2234.16	-1.30	6.12	0.00	2374.61
-1.69	6.02	0.00	2438.50	-1.81	5.80	0.00	2488.24
-2.18	5.92	0.00	2547.25	-2.43	5.87	0.00	2894.48
-8.42	20.97	0.00	42729.25	-9.31	21.08	0.00	41454.94
-9.18	22.07	0.00	37375.25	-7.70	22.22	0.00	36005.85
-6.86	21.48	0.00	38240.64	-6.56	21.72	0.00	40608.57
-6.75	21.10	0.00	43347.39	-7.36	21.40	0.00	48026.18
-7.70	21.46	0.00	54996.84	-8.23	21.14	0.00	57079.48
-8.06	20.42	0.00	46171.79	-8.07	21.02	0.24	45838.94
-7.18	23.38	0.23	40637.10	-6.74	22.67	-0.01	40912.89
-6.92	21.76	-0.01	43670.71	-6.86	21.97	-0.11	44374.44
-6.92	21.06	-0.26	49233.92	-8.37	19.94	0.11	54121.94
-9.66	21.60	-0.08	51154.90	-8.98	21.05	0.00	56784.67
-8.42	19.76	0.00	48682.36	-8.09	22.27	0.31	43185.71
-7.10	24.01	-0.02	39610.05	-6.85	22.91	-0.23	43328.36
-6.69	21.74	-0.12	46409.53	-7.12	21.73	-0.10	48054.71
-7.59	21.08	-0.40	51430.68	-7.92	18.84	-0.14	59922.89
-9.21	20.59	0.54	56822.71	-9.72	22.44	0.00	48007.17
-7.32	19.74	0.00	50070.79	-8.00	22.08	0.51	42681.70
-8.18	25.15	-0.03	39648.09	-7.05	23.42	-0.39	43471.01
-6.99	21.77	-0.29	48397.06	-7.11	21.34	-0.18	50707.94
-7.75	20.59	-0.31	53808.12	-8.10	18.85	-0.13	59114.57
-8.49	18.89	0.69	55329.68	-8.79	24.10	0.00	41483.47
-6.77	21.19	0.00	50222.94	-5.95	20.81	0.46	50251.47
-7.97	24.07	0.54	39410.34	-9.18	24.78	-0.31	39999.95
-7.48	22.56	-0.55	46010.11	-7.37	20.80	-0.33	50451.18
-7.91	20.04	-0.20	52876.17	-7.98	18.88	-0.21	55206.05
-7.66	18.78	0.36	56385.27	-7.87	23.51	0.00	46276.39
-7.63	22.27	0.00	43442.48	-6.53	21.76	-0.33	44754.82
-5.92	21.40	0.21	49081.77	-8.05	23.77	0.53	40237.70
-9.96	23.77	-0.23	39248.68	-8.29	21.50	-0.62	46856.48
-7.70	19.47	-0.29	52885.68	-7.82	19.29	-0.14	54312.14
-7.85	18.73	0.31	55938.30	-8.22	22.84	0.00	46723.34
-7.30	22.58	0.00	41426.41	-7.87	22.24	-0.07	42520.04
-7.35	22.41	-0.40	41749.74	-6.54	21.41	-0.23	49576.27
-7.73	22.66	0.21	47883.54	-8.78	22.65	-0.24	45106.69

-8.02	19.74	-0.44	53266.06	-7.41	19.55	-0.19	54074.39
-7.62	18.44	0.32	57184.08	-7.95	22.55	0.00	45829.43
-7.75	22.59	0.00	40580.05	-7.59	21.99	-0.03	40779.75
-8.30	22.68	-0.01	40503.96	-9.08	21.60	-0.13	40722.69
-8.54	21.83	-0.21	41340.83	-9.11	21.97	-0.23	44678.75
-9.05	20.93	-0.39	47027.66	-8.16	19.64	-0.38	52200.97
-7.41	18.87	0.03	55700.56	-7.39	21.56	0.00	49633.33
-7.49	22.21	0.00	38896.82	-8.16	22.04	-0.10	39904.85
-8.42	21.79	-0.12	40675.14	-8.39	21.84	-0.15	42501.01
-9.05	21.48	-0.06	44431.50	-9.44	21.35	-0.07	44792.87
-9.47	21.45	-0.25	46866.00	-9.07	20.23	-0.28	51383.13
-7.77	20.25	-0.17	52695.48	-7.49	20.43	0.00	53456.26
-7.72	21.63	0.00	38602.02	-8.44	21.36	0.00	38858.78
-9.16	21.62	0.00	39125.05	-9.85	21.44	0.00	40465.92
-9.89	21.11	0.00	43566.11	-9.83	21.34	0.00	46304.92
-9.93	21.24	0.00	47550.70	-9.63	20.92	0.00	48520.69
-9.29	21.15	0.00	49671.38	-8.51	20.64	0.00	56442.32

SECOND SOLUTION FILE (LU=22) FOR SAMPLE PROBLEM 2

10 0 2 180000. 0.6

431.67	-197.99	0.00	2191.24	455.09	-212.12	0.00	2125.89
450.80	-224.79	0.00	1916.68	441.34	-227.25	0.00	1846.45
449.31	-230.96	0.00	1961.06	456.91	-241.77	0.00	2082.49
472.17	-231.50	0.00	2222.94	472.64	-260.84	0.00	2462.88
461.54	-252.21	0.00	2320.35	456.59	-240.91	0.00	2927.15
423.02	-197.25	0.00	2367.78	415.40	-219.66	-5.30	2350.71
391.07	-240.43	-5.43	2083.95	398.76	-233.18	-6.34	2098.10
418.07	-240.56	-7.18	2239.52	425.36	-248.43	-4.98	2275.61
432.90	-246.00	-3.03	2524.82	472.77	-248.69	-13.18	2775.48
483.04	-282.44	-0.42	2623.33	465.69	-252.38	0.00	2912.03
444.29	-199.85	0.00	2496.53	425.42	-237.29	-1.79	2214.65
395.93	-246.46	1.18	2031.28	394.07	-246.80	1.35	2221.97
401.37	-237.37	-2.46	2379.98	414.85	-253.43	-4.16	2464.34
430.10	-255.79	2.69	2637.47	446.97	-234.52	-2.45	3072.97
457.77	-277.62	-13.38	2913.99	456.57	-267.34	0.00	2461.91
439.38	-195.80	0.00	2567.73	437.60	-235.11	-13.98	2188.81
409.21	-290.56	0.69	2033.24	391.49	-258.57	9.29	2229.28
401.08	-246.58	5.27	2481.90	407.66	-248.00	0.79	2600.41
428.61	-253.85	4.13	2759.39	451.92	-235.31	7.06	3031.52
471.29	-226.34	-11.74	2837.42	444.77	-296.35	0.00	2127.36
402.16	-235.79	0.00	2575.54	384.12	-209.36	-8.99	2577.00
426.74	-264.95	-16.35	2021.04	440.74	-304.92	7.59	2051.28
410.37	-257.95	17.51	2359.49	425.36	-236.40	9.33	2587.24
446.74	-237.96	4.46	2711.60	463.24	-221.20	8.49	2831.08
452.54	-217.17	-7.04	2891.55	428.30	-294.92	0.00	2373.15
432.49	-262.15	0.00	2227.82	401.98	-223.99	12.05	2295.12
375.66	-216.51	-5.52	2517.01	431.37	-261.71	-16.55	2063.47
489.06	-288.47	1.93	2012.75	450.90	-248.54	15.29	2402.90
453.13	-218.38	4.64	2712.09	458.47	-221.83	3.54	2785.24
464.23	-219.34	-6.31	2868.63	447.10	-272.47	0.00	2396.07
410.87	-254.29	0.00	2124.43	431.67	-248.09	4.62	2180.51
422.22	-231.15	13.66	2141.01	386.47	-223.48	3.23	2542.37
402.13	-262.71	-5.30	2455.57	450.40	-267.29	5.98	2313.16
454.29	-226.22	13.29	2731.59	447.37	-215.55	5.31	2773.05
465.86	-206.72	-5.42	2932.52	446.44	-262.10	0.00	2350.23
420.95	-249.32	0.00	2081.03	427.40	-227.76	2.84	2091.27
441.76	-239.63	1.16	2077.13	482.52	-227.61	8.39	2088.34
474.53	-218.14	6.25	2120.04	472.17	-248.67	3.89	2291.22
486.56	-235.94	12.44	2411.68	475.88	-215.72	9.08	2676.97
461.52	-202.12	3.88	2856.44	447.01	-245.83	0.00	2545.30
416.56	-228.58	0.00	1994.71	436.57	-222.41	3.04	2046.40
447.29	-218.22	2.95	2085.90	445.90	-215.32	2.43	2179.54
465.94	-226.81	-1.21	2278.54	487.86	-228.03	-0.42	2297.07
488.70	-242.57	3.49	2403.38	487.39	-230.06	8.26	2635.03
468.63	-214.73	8.78	2702.33	462.50	-227.75	0.00	2741.35
420.91	-201.68	0.00	1979.59	447.94	-202.21	0.00	1992.76
466.93	-204.34	0.00	2006.41	484.12	-211.12	0.00	2075.18
488.68	-206.86	0.00	2234.16	488.93	-225.47	0.00	2374.61
499.59	-234.75	0.00	2438.50	505.06	-231.20	0.00	2488.24
495.41	-227.33	0.00	2547.25	485.76	-233.19	0.00	2894.48
1010.39	-469.43	0.00	162729.25	1059.78	-488.55	0.00	161454.94
1074.77	-512.01	0.00	157375.25	1060.97	-511.23	0.00	156005.86
1049.38	-504.03	0.00	158240.64	1037.47	-512.94	0.00	160608.58
1040.29	-490.89	0.00	163347.39	1016.28	-536.19	0.00	168026.19
975.17	-515.07	0.00	174996.84	973.10	-497.21	0.00	177079.48
971.07	-449.11	0.00	166171.80	957.03	-484.54	-8.28	165838.94
939.31	-526.38	-8.61	160637.11	945.74	-506.12	-10.37	160912.89
957.50	-507.28	-11.50	163670.72	960.55	-515.85	-8.24	164374.44
947.83	-501.20	-5.48	169233.92	996.02	-498.89	-19.70	174121.94
1028.68	-560.58	-0.81	171154.91	987.51	-497.00	0.00	176784.67

993.77	-444.21	0.00	168682.36	983.62	-518.27	-2.75	163185.72
948.34	-537.95	1.91	159610.05	919.18	-523.44	1.73	163328.36
911.59	-495.55	-4.12	166409.53	923.38	-515.76	-6.74	168054.72
933.29	-510.94	2.85	171430.69	933.08	-460.48	-3.98	179922.89
964.11	-534.26	-18.70	176822.72	1000.03	-523.75	0.00	168007.17
977.31	-433.51	0.00	170070.80	1004.37	-515.45	-22.43	162681.70
966.51	-615.45	1.09	159648.09	909.75	-544.19	14.16	163471.02
899.24	-507.99	7.48	168397.06	897.50	-501.51	0.73	170707.94
919.64	-502.07	5.26	173808.13	941.59	-460.75	9.96	179114.58
990.59	-446.39	-15.72	175329.69	1008.02	-579.90	0.00	161483.47
910.05	-504.89	0.00	170222.94	872.49	-456.47	-13.81	170251.47
995.45	-574.58	-26.53	159410.34	1010.58	-636.92	11.97	159999.95
922.40	-534.94	26.27	166010.11	926.61	-484.39	13.38	170451.19
951.34	-479.20	6.12	172876.17	971.47	-443.62	11.98	175206.05
951.01	-430.06	-9.47	176385.27	950.51	-570.27	0.00	166276.39
993.49	-568.77	0.00	163442.48	926.35	-496.95	19.18	164754.83
854.81	-471.50	-8.43	169081.77	989.83	-566.41	-26.41	160237.70
1091.03	-609.82	2.79	159248.69	982.79	-516.89	22.39	166856.48
960.38	-449.91	6.27	172885.69	963.81	-449.31	4.91	174312.14
968.70	-439.24	-8.55	175938.30	976.46	-537.25	0.00	166723.34
965.97	-566.62	0.00	161426.41	987.78	-550.73	7.46	162520.05
968.29	-519.67	22.02	161749.73	864.26	-483.15	4.84	169576.27
893.82	-549.54	-7.95	167883.55	986.93	-557.95	8.95	165106.69
958.62	-467.39	18.98	173266.06	943.92	-444.29	7.49	174074.39
965.28	-421.83	-7.29	177184.08	976.65	-530.74	0.00	165829.44
987.11	-569.32	0.00	160580.05	987.68	-528.42	4.64	160779.75
1006.23	-545.81	1.88	160503.97	1069.30	-519.28	13.53	160722.69
1047.72	-495.64	9.91	161340.83	1023.20	-534.10	5.87	164678.75
1035.72	-502.87	18.58	167027.66	996.87	-454.41	12.97	172200.97
961.07	-423.17	5.72	175700.56	960.10	-505.20	0.00	169633.33
988.40	-549.66	0.00	158896.83	1005.85	-532.58	5.11	159904.86
1010.80	-518.15	4.89	160675.14	992.35	-501.91	3.95	162501.02
1013.14	-510.47	-1.88	164431.50	1046.55	-506.37	-0.67	164792.88
1037.36	-521.30	5.20	160866.00	1017.09	-487.17	12.10	171383.13
982.25	-456.28	12.84	172695.48	971.21	-475.84	0.00	173456.27
995.34	-519.19	0.00	158602.02	1027.16	-513.94	0.00	158858.78
1048.89	-510.12	0.00	159125.05	1064.26	-510.54	0.00	160465.92
1051.93	-487.49	0.00	163566.11	1037.11	-506.32	0.00	166304.92
1049.18	-514.01	0.00	167550.70	1055.69	-501.59	0.00	168520.69
1037.43	-489.92	0.00	169671.38	1000.47	-486.26	0.00	176442.33
10	10	2	180000.	0.6			
-35.74	6.33	0.00	2191.24	-37.30	7.52	0.00	2125.89
-36.89	8.28	0.00	1910.68	-37.08	8.32	0.00	1846.45
-38.40	9.21	0.00	1961.06	-39.28	9.78	0.00	2082.49
-40.52	9.41	0.00	2222.94	-40.03	11.81	0.00	2462.88
-38.84	10.78	0.00	2820.35	-38.35	9.87	0.00	2927.15
-35.07	6.26	0.00	2367.78	-34.27	7.95	0.29	2350.71
-32.70	8.36	0.32	2083.95	-33.74	8.24	0.60	2098.10
-35.52	9.31	0.69	2239.52	-36.26	10.09	0.56	2275.61
-36.88	10.22	0.53	2524.82	-39.61	11.30	1.16	2775.48
-39.56	13.36	0.11	2623.33	-38.63	10.37	0.00	2912.03
-36.75	6.79	0.00	2496.53	-35.31	8.77	-0.10	2214.65
-33.24	8.58	-0.09	2031.28	-33.43	9.05	0.07	2221.97
-34.25	8.95	0.34	2379.98	-35.26	10.32	0.49	2464.34
-36.40	11.06	0.12	2637.47	-37.58	10.29	0.36	3072.97
-37.59	13.14	0.77	2913.99	-36.92	10.84	0.00	2461.91
-37.18	6.26	0.00	2567.73	-36.49	8.70	0.88	2188.81
-33.87	11.87	-0.04	2033.24	-33.00	9.83	-0.53	2229.28
-34.16	9.56	-0.23	2481.90	-34.72	10.05	0.09	2600.41
-36.12	11.00	-0.10	2759.39	-38.10	10.41	-0.54	3031.52
-39.42	9.58	0.47	2837.42	-36.44	12.78	0.00	2127.36

-33.88	8.96	0.00	2575.54	-32.93	6.65	0.45	2577.00
-35.56	10.22	1.08	2021.04	-36.29	13.51	-0.45	2051.28
-34.55	10.31	-1.16	2359.49	-36.15	9.34	-0.58	2587.24
-37.86	10.09	-0.23	2711.60	-39.10	9.30	-0.60	2831.08
-38.21	8.86	0.33	2891.55	-35.29	13.10	0.00	2373.15
-36.17	11.13	0.00	2227.82	-34.03	7.66	-0.85	2295.12
-32.14	6.96	0.34	2517.01	-35.83	10.14	1.11	2063.47
-40.20	12.97	0.01	2012.75	-37.72	10.36	-0.90	2402.90
-38.22	8.75	-0.17	2712.09	-38.73	9.22	-0.21	2785.24
-38.87	9.40	0.31	2868.63	-36.66	11.82	0.00	2396.07
-34.18	10.39	0.00	2124.43	-35.77	9.91	-0.38	2180.51
-35.13	8.17	-0.95	2141.01	-32.48	7.80	-0.10	2542.37
-32.91	10.76	0.32	2455.57	-36.82	11.60	-0.35	2313.16
-37.89	9.38	-0.86	2731.59	-37.44	8.59	-0.33	2773.05
-39.08	8.52	0.21	2932.52	-36.65	11.52	0.00	2350.23
-34.78	10.25	0.00	2081.03	-35.27	8.51	-0.24	2091.27
-36.05	9.18	-0.10	2077.13	-39.16	8.85	-0.67	2088.34
-38.92	7.94	-0.42	2120.04	-38.20	10.45	-0.17	2291.22
-39.58	10.08	-0.83	2411.68	-39.45	8.77	-0.52	2676.97
-38.50	7.99	-0.39	2856.44	-37.06	10.63	0.00	2545.30
-34.41	9.10	0.00	1994.71	-35.56	8.49	-0.21	2046.40
-36.06	8.13	-0.18	2085.90	-35.89	7.78	-0.11	2179.54
-37.19	8.91	0.16	2278.54	-39.03	9.33	0.10	2297.07
-39.20	10.35	-0.12	2403.38	-39.38	9.91	-0.53	2635.03
-38.77	8.36	-0.67	2702.33	-38.47	9.55	0.00	2741.35
-34.60	7.57	0.00	1979.59	-36.22	7.67	0.00	1992.76
-37.19	7.65	0.00	2006.41	-38.07	8.15	0.00	2075.18
-38.48	7.85	0.00	2234.16	-38.58	9.00	0.00	2374.61
-39.58	10.02	0.00	2438.50	-40.48	9.68	0.00	2488.24
-39.94	9.07	0.00	2547.25	-39.92	9.62	0.00	2894.48
-91.42	17.27	0.00	162729.25	-95.22	19.02	0.00	161454.94
-97.00	19.91	0.00	157375.25	-97.85	19.59	0.00	156005.86
-97.98	19.93	0.00	158240.64	-97.29	20.51	0.00	160608.58
-97.43	19.19	0.00	163347.39	-94.21	23.49	0.00	168026.19
-89.64	21.25	0.00	174996.84	-88.80	19.90	0.00	177079.48
-87.84	16.00	0.00	166171.80	-86.44	18.69	0.49	165838.94
-85.99	19.44	0.55	160637.11	-87.37	18.38	1.08	160912.89
-88.41	19.84	1.21	163670.72	-88.90	20.41	1.01	164374.44
-87.63	20.26	0.96	169233.92	-90.63	21.71	1.88	174121.94
-92.26	25.59	0.20	171154.91	-89.14	19.80	0.00	176784.67
-89.57	16.47	0.00	168682.36	-89.07	20.29	-0.18	163185.72
-86.97	19.68	-0.17	159610.05	-84.43	19.79	0.17	163328.36
-84.00	18.65	0.60	166409.53	-84.65	20.73	0.85	168054.72
-85.09	21.20	0.31	171430.69	-84.72	19.31	0.62	179922.89
-86.18	24.27	1.13	176822.72	-89.29	20.26	0.00	168007.17
-89.42	15.40	0.00	170070.80	-91.25	20.28	1.54	162681.70
-87.18	25.96	-0.06	159648.09	-83.12	21.16	-0.87	163471.02
-82.24	19.85	-0.34	168397.06	-81.97	19.83	0.19	170707.94
-83.38	20.96	-0.08	173808.13	-85.25	19.26	-0.83	179114.58
-89.91	17.60	0.59	175329.69	-91.40	23.37	0.00	161483.47
-83.25	20.58	0.00	170222.94	-80.68	16.10	0.74	170251.47
-90.39	23.34	1.92	159410.34	-90.27	28.65	-0.77	159999.95
-83.84	21.43	-1.89	166010.11	-84.53	18.83	-0.89	170451.19
-86.34	19.39	-0.33	172876.17	-88.42	17.39	-0.92	175206.05
-86.89	16.02	0.43	176385.27	-86.67	23.23	0.00	166276.39
-90.42	25.56	0.00	163442.48	-85.26	18.85	-1.47	164754.83
-78.94	16.69	0.55	169081.77	-89.76	22.88	1.93	160237.70
-97.43	27.32	0.06	159248.69	-88.88	21.10	-1.38	166856.48
-87.50	17.19	-0.21	172885.69	-87.74	17.32	-0.30	174312.14
-88.29	17.02	0.42	175938.30	-88.66	20.77	0.00	166723.34
-87.97	24.88	0.00	161426.41	-89.48	23.66	-0.66	162520.05

-88.38	20.13	-1.06	161749.73	-79.03	17.82	-0.15	169576.27
-80.41	22.74	0.51	167883.55	-88.49	23.58	-0.56	165106.69
-86.76	18.56	-1.30	173266.06	-86.18	16.40	-0.49	174074.39
-88.14	15.66	0.27	177184.08	-88.92	20.55	0.00	165829.44
-89.36	25.12	0.00	160580.05	-89.79	21.71	-0.42	160779.75
-90.78	22.40	-0.17	160503.97	-96.24	21.23	-1.19	160722.69
-94.94	18.42	-0.71	161340.83	-91.66	22.13	-0.26	164678.75
-93.07	20.42	-1.33	167027.66	-90.43	17.31	-0.77	172200.97
-87.91	15.17	-0.64	175700.56	-87.91	19.46	0.00	169633.33
-89.74	23.60	0.00	158890.83	-90.73	22.01	-0.38	159904.86
-91.01	20.85	-0.32	160675.14	-89.32	19.07	-0.18	162501.02
-90.58	20.45	0.28	164431.50	-93.51	20.20	0.18	164792.88
-92.55	21.55	-0.16	166866.00	-91.08	19.81	-0.82	171383.13
-89.45	16.56	-1.07	172695.43	-88.78	18.22	0.00	173456.27
-90.01	21.23	0.00	158602.02	-92.43	21.01	0.00	158858.78
-93.79	20.23	0.00	159125.05	-94.49	20.53	0.00	160465.92
-93.26	18.64	0.00	163566.11	-91.92	20.21	0.00	166304.92
-93.04	21.12	0.00	167550.70	-94.15	20.29	0.00	168520.69
-92.81	18.73	0.00	169671.38	-90.17	19.10	0.00	176442.33

10 9 2 180000. 0.6

6.32	-28.95	0.00	2191.24	8.27	-28.26	0.00	2125.89
8.92	-28.64	0.00	1916.68	6.47	-27.92	0.00	1846.45
4.42	-26.90	0.00	1961.06	3.22	-27.82	0.00	2082.49
2.32	-27.77	0.00	2222.94	2.35	-28.80	0.00	2462.88
2.00	-31.13	0.00	2820.35	2.24	-31.45	0.00	2927.15
5.92	-28.34	0.00	2367.78	6.54	-29.07	-0.46	2350.71
6.10	-31.96	-0.41	2083.95	5.20	-30.35	0.10	2098.10
4.94	-28.58	0.13	2239.52	4.66	-29.18	0.30	2275.61
3.66	-28.88	0.65	2524.82	4.44	-27.60	0.03	2775.48
6.31	-30.09	0.19	2623.33	4.11	-31.51	0.00	2912.03
6.13	-27.23	0.00	2496.53	6.73	-30.44	-0.66	2214.65
5.96	-32.75	0.03	2031.28	5.40	-30.97	0.47	2221.97
4.79	-29.27	0.29	2379.98	5.18	-29.38	0.29	2464.34
5.31	-29.12	0.86	2637.47	4.34	-26.90	0.40	3072.97
6.24	-29.45	-1.02	2913.99	7.06	-33.08	0.00	2461.91
4.06	-27.59	0.00	2567.73	6.34	-29.79	-0.93	2188.81
8.27	-33.81	0.06	2033.24	6.05	-31.68	0.72	2229.28
5.22	-29.60	0.57	2481.90	5.24	-29.29	0.39	2600.41
5.64	-28.70	0.64	2759.39	4.89	-26.86	0.18	3031.52
4.51	-27.38	-1.42	2837.42	6.22	-34.72	0.00	2127.36
4.38	-29.56	0.00	2575.54	3.23	-29.87	-0.92	2577.00
7.36	-31.98	-0.94	2021.04	9.63	-32.26	0.55	2051.28
6.27	-30.38	0.95	2359.49	5.10	-28.57	0.59	2587.24
5.51	-27.82	0.39	2711.60	4.71	-26.87	0.34	2831.08
3.78	-27.68	-0.73	2891.55	5.15	-34.64	0.00	2373.15
6.19	-29.17	0.00	2227.82	4.58	-29.93	0.58	2295.12
3.74	-30.64	-0.40	2517.01	7.39	-31.50	-0.91	2063.47
9.79	-30.27	0.44	2012.75	6.59	-28.99	1.13	2402.90
4.83	-27.29	0.58	2712.09	.83	-27.70	0.26	2785.24
4.41	-27.30	-0.62	2868.63	.91	-33.91	0.00	2396.07
6.79	-29.52	0.00	2124.43	7.09	-29.62	0.10	2180.51
6.18	-30.30	0.70	2141.01	4.74	-30.50	0.49	2542.37
6.92	-31.51	-0.38	2455.57	7.85	-30.83	0.45	2313.16
5.53	-27.88	0.80	2731.59	4.49	-28.43	0.34	2773.05
4.12	-27.18	-0.67	2932.52	5.99	-33.09	0.00	2350.23
7.89	-29.60	0.00	2081.03	7.05	-29.28	0.05	2091.27
7.83	-30.47	0.02	2077.13	7.97	-28.37	0.20	2088.34
6.72	-29.46	0.36	2120.04	7.76	-29.96	0.44	2291.22
7.20	-28.87	0.69	2411.68	5.38	-28.19	0.73	2676.97
4.08	-27.84	-0.14	2356.44	4.81	-31.87	0.00	2545.30
8.13	-29.19	0.00	1994.71	8.22	-29.60	0.18	2046.40

7.97	-29.52	0.23	2085.90	7.37	-30.18	0.29	2179.54
7.82	-29.61	0.14	2278.54	7.96	-29.35	0.16	2297.07
7.97	-29.86	0.50	2403.38	7.19	-28.71	0.52	2635.03
4.95	-29.80	0.26	2702.33	4.54	-30.19	0.00	2741.35
8.83	-29.32	0.00	1979.59	8.71	-28.94	0.00	1992.76
9.03	-29.32	0.00	2006.41	9.50	-29.16	0.00	2075.18
8.96	-29.24	0.00	2234.16	8.57	-29.97	0.00	2374.61
8.60	-29.71	0.00	2438.50	8.01	-29.36	0.00	2488.24
7.84	-30.45	0.00	2547.25	6.24	-30.62	0.00	2894.48
-25.06	-74.85	0.00	162729.25	-23.95	-74.27	0.00	161454.94
-25.77	-77.85	0.00	157375.25	-32.62	-78.70	0.00	156005.86
-36.42	-75.48	0.00	158240.64	-37.52	-76.08	0.00	160608.58
-37.26	-74.37	0.00	163347.39	-33.15	-72.63	0.00	168026.19
-29.22	-74.35	0.00	174996.84	-26.90	-73.94	0.00	177079.48
-24.03	-73.35	0.00	166171.80	-23.21	-74.27	-0.69	165838.94
-26.70	-83.59	-0.58	160637.11	-29.60	-81.44	0.76	160912.89
-29.83	-76.76	0.86	163670.72	-30.66	-77.25	1.12	164374.44
-29.98	-73.61	1.73	169233.92	-26.57	-68.04	0.82	174121.94
-22.86	-72.38	0.46	171154.91	-23.96	-74.09	0.00	176784.67
-23.47	-70.34	0.00	168682.36	-24.66	-78.38	-1.41	163185.72
-27.50	-80.09	-0.03	159610.05	-27.13	-81.55	1.06	163328.36
-27.60	-77.51	0.88	166409.53	-26.69	-76.14	0.99	168054.72
-25.38	-73.16	1.86	171430.69	-24.25	-65.25	0.98	179922.89
-20.43	-69.34	-1.46	176822.72	-20.87	-80.08	0.00	168007.17
-27.67	-70.96	0.00	170070.80	-26.20	-77.61	-1.12	162681.70
-23.01	-86.70	0.10	159648.09	-25.32	-82.81	1.06	163471.02
-25.28	-76.95	0.99	168397.06	-24.87	-75.18	0.86	170707.94
-23.40	-71.43	1.23	173808.13	-23.60	-65.48	-0.01	179114.58
-25.65	-67.08	-2.47	175329.69	-25.85	-85.40	0.00	161483.47
-25.68	-73.61	0.00	170222.94	-27.62	-75.18	-1.41	170251.47
-25.79	-84.04	-0.98	159410.34	-21.04	-83.62	0.76	159999.95
-24.16	-79.25	1.02	166010.11	-25.34	-73.75	0.77	170451.19
-24.50	-70.32	0.62	172876.17	-25.90	-67.02	0.26	175206.05
-26.60	-67.84	-1.21	176385.27	-26.04	-83.17	0.00	166276.39
-26.41	-74.78	0.00	163442.48	-27.93	-77.32	0.42	164754.83
-26.66	-77.47	-0.52	169081.77	-25.21	-83.32	-0.92	160237.70
-22.61	-80.48	1.00	159248.69	-24.19	-75.24	1.64	166856.48
-25.89	-69.53	1.05	172885.69	-25.76	-68.90	0.37	174312.14
-26.33	-67.06	-1.02	175938.30	-25.44	-82.22	0.00	166723.34
-25.83	-76.57	0.00	161426.41	-24.83	-76.23	-0.14	162520.05
-26.61	-79.46	0.58	161749.73	-24.29	-76.98	0.87	169576.27
-20.55	-79.02	-0.52	167883.55	-21.68	-78.51	0.65	165106.69
-23.85	-69.86	0.97	173266.06	-26.08	-70.68	0.45	174074.39
-26.76	-66.80	-1.16	177184.03	-26.27	-81.17	0.00	165829.44
-24.51	-76.58	0.00	160580.05	-25.89	-76.59	-0.13	160779.75
-24.02	-79.22	-0.06	160503.97	-24.76	-75.63	-0.21	160722.69
-26.47	-78.52	0.39	161340.83	-22.19	-76.75	0.77	164678.75
-23.46	-73.65	0.75	167027.66	-25.51	-70.49	1.08	172200.97
-27.14	-68.90	-0.57	175700.56	-27.48	-77.64	0.00	169633.33
-25.42	-76.24	0.00	158896.83	-23.89	-76.84	0.16	159904.86
-23.51	-76.69	0.30	160675.14	-22.99	-78.16	0.51	162501.02
-21.39	-75.83	0.42	164431.50	-21.89	-75.52	0.41	164792.88
-21.20	-75.11	0.94	166866.00	-21.95	-71.30	0.64	171383.13
-26.35	-73.64	0.01	172695.48	-27.26	-73.55	0.00	173456.27
-24.34	-75.82	0.00	158602.02	-23.66	-75.03	0.00	158858.78
-22.16	-76.60	0.00	159125.05	-20.18	-75.66	0.00	160465.92
-19.58	-75.54	0.00	163566.11	-19.15	-75.50	0.00	166304.92
-19.52	-74.55	0.00	167550.70	-21.48	-73.84	0.00	168520.69
-22.11	-75.87	0.00	169671.38	-23.63	-73.55	0.00	176442.33
10 6	2 160000.	0.6					
27.96	-9.84	0.00	2191.24	29.20	-10.50	0.00	2125.89

31.01	-11.77	0.00	1916.68	32.58	-12.22	0.00	1846.45
32.95	-11.93	0.00	1961.06	32.76	-11.19	0.00	2082.49
32.11	-10.34	0.00	2222.94	29.73	-9.85	0.00	2462.88
26.63	-7.65	0.00	2820.35	25.05	-6.74	0.00	2927.15
26.38	-8.87	0.00	2367.78	26.52	-9.18	-0.26	2350.71
28.14	-9.97	-0.31	2083.95	29.25	-10.01	-0.49	2098.10
29.70	-9.81	-0.55	2239.52	30.12	-9.73	-0.45	2275.61
28.97	-8.61	-0.40	2524.82	27.80	-8.25	-0.83	2775.48
27.04	-8.79	-0.07	2623.33	24.88	-6.20	0.00	2912.03
25.94	-8.53	0.00	2496.53	27.40	-9.73	0.00	2214.65
28.53	-10.00	0.08	2031.28	28.22	-9.22	-0.02	2221.97
28.37	-8.65	-0.25	2379.98	28.65	-8.52	-0.38	2464.34
28.12	-7.99	-0.10	2637.47	26.08	-6.55	-0.27	3072.97
25.04	-7.24	-0.48	2913.99	25.35	-6.95	0.00	2461.91
25.62	-7.79	0.00	2567.73	27.67	-9.61	-0.76	2188.81
28.13	-10.88	0.04	2033.24	27.42	-9.25	0.47	2229.28
27.34	-7.95	0.19	2431.90	27.58	-7.59	-0.07	2600.41
27.23	-7.21	0.05	2759.39	26.26	-6.22	0.36	3031.52
26.34	-6.11	-0.25	2837.42	27.12	-8.16	0.00	2127.36
23.79	-8.21	0.00	2575.54	24.21	-6.85	-0.35	2577.00
28.29	-10.35	-0.97	2021.04	29.02	-11.24	0.42	2051.28
27.50	-8.52	0.92	2359.49	27.49	-7.10	0.42	2587.24
27.79	-6.85	0.16	2711.60	27.26	-6.07	0.40	2831.08
25.57	-5.19	-0.17	2891.55	25.14	-6.82	0.00	2373.15
25.40	-10.42	0.00	2227.82	25.19	-8.15	0.64	2295.12
24.24	-6.70	-0.25	2517.01	28.05	-9.82	-0.96	2063.47
30.57	-11.03	0.04	2012.75	28.38	-8.01	0.69	2402.90
27.16	-6.25	0.10	2712.09	26.67	-5.74	0.13	2785.24
25.44	-5.39	-0.16	2868.63	25.13	-6.12	0.00	2396.07
24.47	-10.39	0.00	2124.43	25.25	-9.29	0.27	2180.51
25.97	-8.68	0.75	2141.01	23.89	-6.73	0.08	2542.37
24.46	-7.73	-0.24	2455.57	26.87	-8.43	0.28	2313.16
26.16	-6.19	0.56	2731.69	25.67	-5.32	0.21	2773.05
24.91	-4.79	-0.09	2932.52	25.21	-6.30	0.00	2350.23
23.96	-10.31	0.00	2081.03	24.74	-9.10	0.17	2091.27
25.61	-9.09	0.07	2077.13	27.21	-9.33	0.50	2088.34
27.91	-8.53	0.34	2120.04	26.93	-8.11	0.14	2291.22
26.85	-7.36	0.60	2411.68	25.89	-5.70	0.32	2676.97
24.60	-4.66	0.26	2856.44	24.40	-5.49	0.00	2545.30
23.51	-10.16	0.00	1994.71	24.16	-9.02	0.16	2046.40
24.64	-8.60	0.15	2085.90	24.75	-7.94	0.10	2179.54
25.09	-7.79	-0.10	2278.54	26.07	-7.80	-0.06	2297.07
25.96	-7.24	0.09	2403.38	24.96	-6.12	0.34	2635.03
24.54	-4.85	0.42	2702.33	23.80	-4.75	0.00	2741.35
22.92	-9.10	0.00	1979.59	24.16	-8.84	0.00	1992.76
24.87	-8.69	0.00	2006.41	25.19	-8.43	0.00	2075.18
25.06	-7.55	0.00	2234.16	24.86	-6.97	0.00	2374.61
25.09	-6.95	0.00	2438.50	25.29	-6.49	0.00	2488.24
24.55	-5.48	0.00	2547.25	22.87	-4.20	0.00	2894.48
-1.06	37.94	0.00	162729.25	-2.09	37.93	0.00	161454.94
-1.35	39.73	0.00	157375.25	1.90	40.08	0.00	156005.86
3.72	38.61	0.00	158240.64	4.34	39.02	0.00	160608.58
4.13	38.02	0.00	163347.39	2.49	37.88	0.00	168026.19
1.10	38.43	0.00	174996.84	-0.02	38.05	0.00	177079.48
-1.08	37.07	0.00	166171.80	-1.21	37.93	0.40	165838.94
0.66	42.47	0.35	160637.11	1.90	41.30	-0.21	160912.89
1.88	39.32	-0.24	163670.72	2.23	39.65	-0.39	164374.44
2.01	37.92	-0.69	169233.92	-0.21	35.50	-0.13	174121.94
-2.31	38.14	-0.19	171154.91	-1.48	38.14	0.00	176784.67
-1.66	35.72	0.00	168682.36	-0.90	40.12	0.64	163185.72
0.89	43.70	-0.01	159610.05	1.09	41.61	-0.48	163328.36

1.50	39.55	-0.33	166409.53	0.89	39.23	-0.35	168054.72
0.15	37.90	-0.84	171430.69	-0.50	33.88	-0.38	179922.89
-2.63	36.57	0.86	176822.72	-2.90	41.05	0.00	168007.17
0.33	35.87	0.00	170070.80	-0.51	39.75	0.75	162681.70
-1.32	44.95	-0.06	159648.09	0.39	42.43	-0.63	163471.02
0.50	39.50	-0.52	168397.06	0.36	38.69	-0.38	170707.94
-0.59	37.09	-0.60	173808.13	-0.89	34.02	-0.11	179114.58
-0.68	34.52	1.26	175329.69	-0.74	44.06	0.00	161483.47
0.27	37.93	0.00	170222.94	1.63	38.04	0.78	170251.47
-0.50	43.28	0.74	159410.34	-2.71	43.92	-0.47	159999.95
-0.28	40.83	-0.75	166010.11	0.17	37.87	-0.49	170451.19
-0.52	36.35	-0.34	172876.17	-0.23	34.49	-0.26	175206.05
0.22	34.69	0.64	176385.27	-0.01	43.07	0.00	166276.39
-0.42	39.17	0.00	163442.48	1.12	39.43	-0.41	164754.83
1.47	39.25	0.33	169081.77	-0.67	42.92	0.71	160237.70
-2.99	42.25	-0.47	159248.69	-1.02	38.91	-0.98	166856.48
-0.06	35.66	-0.53	172885.69	-0.21	35.41	-0.22	174312.14
-0.12	34.50	0.55	175938.30	-0.61	42.29	0.00	166723.34
-0.34	39.92	0.00	161426.41	-1.01	39.62	-0.03	162520.05
0.09	40.66	-0.51	161749.73	0.33	39.22	-0.44	169576.27
-1.60	40.96	0.32	167883.55	-2.19	40.85	-0.39	165106.69
-0.97	36.04	-0.65	173266.06	0.15	36.14	-0.28	174074.39
0.09	34.19	0.59	177184.08	-0.24	41.77	0.00	165829.44
-1.21	39.98	0.00	160580.05	-0.55	39.51	0.00	160779.75
-1.54	40.89	0.00	160503.97	-1.95	39.02	-0.07	160722.69
-0.89	40.02	-0.29	161340.83	-2.46	39.79	-0.41	164678.75
-2.07	38.09	-0.55	167027.66	-0.77	36.17	-0.63	172200.97
0.30	35.12	0.18	175700.56	0.45	39.95	0.00	169633.33
-0.85	39.61	0.00	158896.83	-1.67	39.71	-0.13	159904.86
-1.84	39.49	-0.19	160675.14	-1.81	39.95	-0.27	162501.02
-2.75	39.08	-0.16	164431.50	-2.92	38.91	-0.17	164792.88
-3.12	38.97	-0.47	166866.00	-2.60	36.93	-0.42	171383.13
-0.30	37.59	-0.16	172695.48	0.16	37.83	0.00	173456.27
-1.44	39.12	0.00	158602.02	-2.04	38.73	0.00	158858.78
-2.90	39.37	0.00	159125.05	-3.92	38.98	0.00	160465.92
-4.03	38.68	0.00	163566.11	-4.04	38.93	0.00	166304.92
-4.03	38.64	0.00	167550.70	-3.29	38.20	0.00	168520.69
-2.84	38.96	0.00	169671.38	-1.86	37.98	0.00	176442.33
10 7	2 180000.	0.6					
-2.66	8.77	0.00	2191.24	-3.64	8.36	0.00	2125.89
-4.26	8.36	0.00	1916.68	-4.41	8.06	0.00	1846.45
-4.99	7.89	0.00	1961.06	-5.79	8.23	0.00	2082.49
-7.09	8.30	0.00	2222.94	-8.68	9.25	0.00	2462.88
-9.91	9.79	0.00	2820.35	-10.89	9.79	0.00	2927.15
-2.07	8.62	0.00	2367.78	-2.73	8.38	0.14	2350.71
-2.95	8.85	0.13	2083.95	-3.63	8.35	0.03	2098.10
-4.67	8.05	0.05	2239.52	-5.59	8.37	0.02	2275.61
-6.64	8.34	-0.04	2524.82	-8.93	8.22	0.25	2775.48
-10.95	9.07	-0.01	2623.33	-11.19	8.70	0.00	2912.03
-1.65	8.25	0.00	2496.53	-2.14	8.60	0.17	2214.65
-2.38	8.79	-0.01	2031.28	-3.32	8.29	-0.11	2221.97
-4.19	7.93	-0.03	2379.98	-5.22	8.18	0.00	2464.34
-6.46	8.22	-0.16	2637.47	-7.81	7.38	-0.01	3072.97
-9.81	8.14	0.39	2913.99	-11.15	7.82	0.00	2461.91
-0.44	8.14	0.00	2567.73	-1.35	8.20	0.24	2188.81
-2.39	9.05	-0.02	2033.24	-2.90	8.29	-0.20	2229.28
-3.82	7.79	-0.15	2481.90	-4.70	7.85	-0.07	2600.41
-5.77	7.78	-0.15	2759.39	-7.07	6.99	-0.12	3031.52
-8.72	6.29	0.37	2837.42	-10.02	7.17	0.00	2127.36
0.50	8.61	0.00	2575.54	-0.06	7.89	0.23	2577.00
-1.53	8.21	0.23	2021.04	-3.21	8.39	-0.14	2051.28

-3.45	7.76	-0.29	2359.49	-4.13	7.27	-0.18	2587.24
-5.11	7.09	-0.10	2711.60	-5.96	6.38	-0.14	2831.08
-6.90	5.78	0.19	2891.55	-8.12	6.71	0.00	2373.15
1.33	8.56	0.00	2227.82	0.76	7.76	-0.11	2295.12
-0.23	7.48	0.10	2517.01	-1.86	7.65	0.22	2063.47
-3.67	7.52	-0.10	2012.75	-3.79	7.07	-0.30	2402.90
-4.02	6.43	-0.13	2712.09	-4.64	6.19	-0.07	2785.24
-5.67	5.51	0.15	2868.63	-7.00	5.87	0.00	2396.07
2.50	8.43	0.00	2124.43	1.18	7.57	0.01	2180.51
0.36	7.13	-0.12	2141.01	-0.68	7.08	-0.11	2542.37
-2.01	7.43	0.09	2455.57	-3.01	7.22	-0.11	2313.16
-3.20	6.30	-0.21	2731.59	-3.26	6.04	-0.09	2773.05
-4.23	5.23	0.15	2932.52	-5.44	5.44	0.00	2350.23
3.41	8.15	0.00	2081.03	2.34	7.07	0.02	2091.27
0.91	6.93	0.00	2077.13	-0.46	6.22	-0.01	2088.34
-1.16	6.35	-0.07	2120.04	-2.20	6.60	-0.10	2291.22
-2.58	6.20	-0.16	2411.68	-2.57	5.78	-0.16	2676.97
-2.91	5.20	0.01	2856.44	-3.85	5.33	0.00	2545.30
4.71	7.33	0.00	1994.71	3.00	6.67	-0.01	2046.40
1.56	6.27	-0.03	2085.90	0.38	6.36	-0.06	2179.54
-0.80	6.27	-0.04	2278.54	-1.56	6.14	-0.04	2297.07
-1.96	6.21	-0.11	2403.38	-2.16	5.83	-0.11	2635.03
-2.08	5.68	-0.06	2702.33	-2.77	5.18	0.00	2741.35
5.50	6.15	0.00	1979.59	3.46	5.61	0.00	1992.76
1.76	5.77	0.00	2006.41	0.36	5.84	0.00	2075.18
-0.65	5.91	0.00	2234.16	-1.30	6.12	0.00	2374.61
-1.69	6.02	0.00	2438.50	-1.81	5.80	0.00	2488.24
-2.18	5.92	0.00	2547.25	-2.43	5.87	0.00	2894.48
-8.42	20.97	0.00	162729.25	-9.31	21.08	0.00	161454.94
-9.18	22.07	0.00	157375.25	-7.70	22.22	0.00	156005.86
-6.86	21.48	0.00	158240.64	-6.56	21.72	0.00	160608.58
-6.75	21.10	0.00	163347.39	-7.36	21.40	0.00	168026.19
-7.70	21.46	0.00	174996.84	-8.23	21.14	0.00	177079.48
-8.06	20.42	0.00	166171.80	-8.07	21.02	0.24	165838.94
-7.18	23.38	0.23	160637.11	-6.74	22.67	-0.01	160912.89
-6.92	21.76	-0.01	163670.72	-6.86	21.97	-0.11	164374.44
-6.92	21.06	-0.26	169233.92	-8.37	19.94	0.11	174121.94
-9.66	21.60	-0.08	171154.91	-8.98	21.05	0.00	176784.67
-8.42	19.76	0.00	168682.36	-8.09	22.27	0.31	163185.72
-7.10	24.01	-0.02	159610.05	-6.85	22.91	-0.23	163328.36
-6.69	21.74	-0.12	166409.53	-7.12	21.73	-0.10	168054.72
-7.59	21.08	-0.40	171430.69	-7.92	18.84	-0.14	179922.89
-9.21	20.59	0.54	176822.72	-9.72	22.44	0.00	168007.17
-7.32	19.74	0.00	170070.80	-8.00	22.08	0.51	162681.70
-8.18	25.15	-0.03	159648.09	-7.05	23.42	-0.39	163471.02
-6.99	21.77	-0.29	168397.06	-7.11	21.34	-0.18	170707.94
-7.75	20.59	-0.31	173808.13	-8.10	18.85	-0.13	179114.58
-8.49	18.89	0.69	175329.69	-8.79	24.10	0.00	161483.47
-6.77	21.19	0.00	170222.94	-5.95	20.81	0.46	170251.47
-7.97	24.07	0.54	159410.34	-9.18	24.78	-0.31	159999.95
-7.48	22.56	-0.55	166010.11	-7.37	20.80	-0.33	170451.19
-7.91	20.04	-0.20	172876.17	-7.98	18.88	-0.21	175206.05
-7.66	18.78	0.36	176385.27	-7.87	23.51	0.00	166276.39
-7.63	22.27	0.00	163442.48	-6.53	21.76	-0.33	164754.83
-5.92	21.40	0.21	169081.77	-8.05	23.77	0.53	160237.70
-9.96	23.77	-0.23	159248.69	-8.29	21.50	-0.62	166856.48
-7.70	19.47	-0.29	172885.69	-7.82	19.29	-0.14	174312.14
-7.85	18.73	0.31	175938.30	-8.22	22.84	0.00	166723.34
-7.30	22.58	0.00	161426.41	-7.87	22.24	-0.07	162520.05
-7.35	22.41	-0.40	161749.73	-6.54	21.41	-0.23	169576.27
-7.73	22.66	0.21	167683.55	-8.78	22.65	-0.24	165106.69

-8.02	19.74	-0.44	173266.06	-7.41	19.55	-0.19	174074.39
-7.62	18.44	0.32	177184.08	-7.95	22.55	0.00	165829.44
-7.75	22.59	0.00	160580.05	-7.59	21.99	-0.03	160779.75
-8.30	22.68	-0.01	160503.97	-9.03	21.60	-0.13	160722.69
-8.54	21.83	-0.21	161340.83	-9.11	21.97	-0.23	164678.75
-9.05	20.93	-0.39	167027.66	-8.16	19.64	-0.38	172200.97
-7.41	18.87	0.03	175700.56	-7.39	21.56	0.00	169633.33
-7.49	22.21	0.00	158896.83	-8.16	22.04	-0.10	159904.86
-8.42	21.79	-0.12	160675.14	-8.39	21.84	-0.15	162501.02
-9.05	21.48	-0.06	164431.50	-9.44	21.35	-0.07	164792.88
-9.47	21.45	-0.25	166366.00	-9.07	20.23	-0.28	171383.13
-7.77	20.25	-0.17	172695.48	-7.49	20.43	0.00	173456.27
-7.72	21.63	0.00	158602.02	-8.44	21.36	0.00	158858.78
-9.16	21.62	0.00	159125.05	-9.85	21.44	0.00	160465.92
-9.89	21.11	0.00	163566.11	-9.83	21.34	0.00	166304.92
-9.93	21.24	0.00	167550.70	-9.63	20.92	0.00	168520.69
-9.29	21.15	0.00	169671.38	-8.51	20.64	0.00	176442.33

INTERACTIVE INPUTS FOR SAMPLE PROBLEM 2

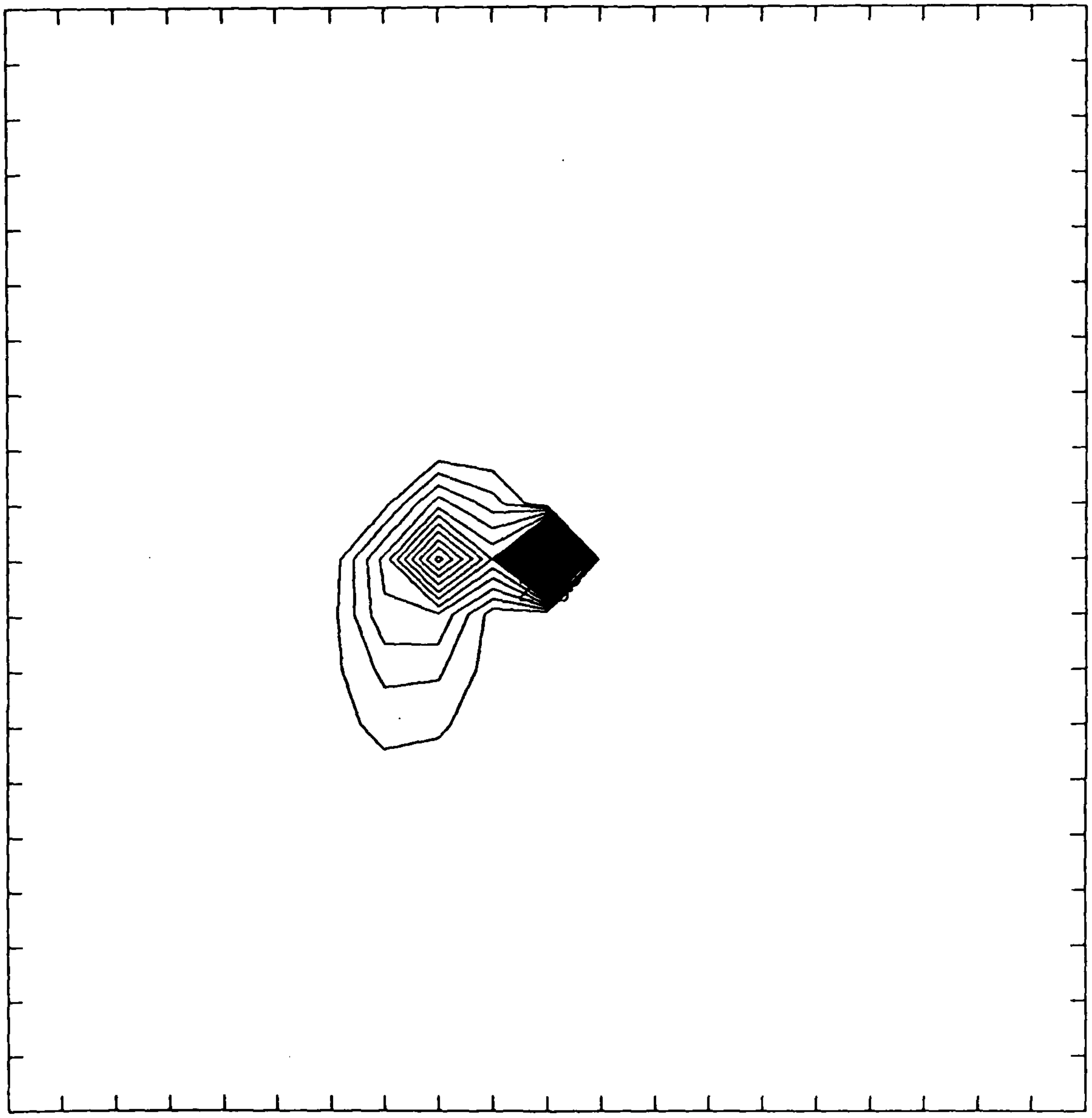
```
RECEPTOR SPACING (M)?
   1000.000
HOW MANY WIND SITES?
         4
HOW MANY EMP. ORTH. FCTNS?
         3
SOURCE X,Y,Z (ABOVE SFC) IN METERS?
  0.0000000E+00  0.0000000E+00   30.00000
HOUR?
         7
FOR HR FROM              7 TO           8
MIX HT. (M) & STABIL.?--NEG. MIX HT TO STOP
   620.0000              1
SOURCE STRNGTH (G/S)?
   1000.000
WIND SPD(M/S) & DIR. AT SITE            1 FOR HR              7
   1.000000       270.0000
WIND SPD(M/S) & DIR. AT SITE            2 FOR HR              7
   1.500000       330.0000
WIND SPD(M/S) & DIR. AT SITE            3 FOR HR              7
   2.100000       300.0000
WIND SPD(M/S) & DIR. AT SITE            4 FOR HR              7
   1.700000       310.0000
GRADIENT WIND SPD. (M/S) & DIR.                  FOR HR.             7?
   5.000000       10.00000
WIND SPD(M/S) & DIR. AT SITE            1 FOR HR              8
   1.000000       90.00000
WIND SPD(M/S) & DIR. AT SITE            2 FOR HR              8
   1.000000       90.00000
WIND SPD(M/S) & DIR. AT SITE            3 FOR HR              8
   1.000000       90.00000
WIND SPD(M/S) & DIR. AT SITE            4 FOR HR              8
   1.000000       90.00000
GRADIENT WIND SPD. (M/S) & DIR.                  FOR HR.             8?
   5.000000       135.0000
FOR HR FROM              8 TO           9
MIX HT. (M) & STABIL.?--NEG. MIX HT TO STOP
   1700.000              4
SOURCE STRNGTH (G/S)?
   1500.000
WIND SPD(M/S) & DIR. AT SITE            1 FOR HR              9
   2.000000       300.0000
WIND SPD(M/S) & DIR. AT SITE            2 FOR HR              9
   2.500000      0.0000000E+00
WIND SPD(M/S) & DIR. AT SITE            3 FOR HR              9
   3.100000       330.0000
WIND SPD(M/S) & DIR. AT SITE            4 FOR HR              9
   2.700000       340.0000
GRADIENT WIND SPD. (M/S) & DIR.                  FOR HR.             9?
   6.500000       40.00000
FOR HR FROM              9 TO          10
MIX HT. (M) & STABIL.?--NEG. MIX HT TO STOP
  -100.0000              3
```

INTERACTIVE OUTPUTS FOR SAMPLE PROBLEM 2

HOUR NO.	0	
-7000.000	-3000.000	1.4041055E-08
-6000.000	-5000.000	5.1520832E-08
-6000.000	-4000.000	2.5518435E-07
-6000.000	-3000.000	5.1387809E-07
-6000.000	-2000.000	4.4097987E-07
-6000.000	-1000.000	1.2672075E-07
-6000.000	0.0000000E+00	1.5517774E-08
-5000.000	-6000.000	5.6939321E-08
-5000.000	-5000.000	7.6661604E-07
-5000.000	-4000.000	3.8304647E-06
-5000.000	-3000.000	8.1432008E-06
-5000.000	-2000.000	7.8245639E-06
-5000.000	-1000.000	3.3463073E-06
-5000.000	0.0000000E+00	6.0919206E-07
-4000.000	-6000.000	3.4446271E-07
-4000.000	-5000.000	4.1964154E-06
-4000.000	-4000.000	2.3577797E-05
-4000.000	-3000.000	5.4558688E-05
-4000.000	-2000.000	6.6677319E-05
-4000.000	-1000.000	5.7781825E-05
-4000.000	0.0000000E+00	2.2884255E-05
-4000.000	1000.000	1.5719984E-06
-3000.000	-7000.000	2.3149786E-08
-3000.000	-6000.000	7.6661604E-07
-3000.000	-5000.000	9.3800772E-06
-3000.000	-4000.000	5.3583644E-05
-3000.000	-3000.000	1.3714723E-04
-3000.000	-2000.000	2.2989418E-04
-3000.000	-1000.000	3.7888405E-04
-3000.000	0.0000000E+00	4.3270690E-04
-3000.000	1000.000	6.5545071E-05
-3000.000	2000.000	2.6286153E-07
-2000.000	-7000.000	1.8953445E-08
-2000.000	-6000.000	6.2765224E-07
-2000.000	-5000.000	8.4874455E-06
-2000.000	-4000.000	4.4082473E-05
-2000.000	-3000.000	1.1838430E-04
-2000.000	-2000.000	2.1884797E-04
-2000.000	-1000.000	3.9335105E-04
-2000.000	0.0000000E+00	1.1431955E-03
-2000.000	1000.000	4.5502241E-04
-2000.000	2000.000	1.0136596E-06
-1000.000	-6000.000	2.3090035E-07
-1000.000	-5000.000	2.8129414E-06
-1000.000	-4000.000	1.5894730E-05
-1000.000	-3000.000	3.8497048E-05
-1000.000	-2000.000	4.9093374E-05
-1000.000	-1000.000	4.8151996E-05
-1000.000	0.0000000E+00	5.9150637E-04
-1000.000	1000.000	2.4708893E-04
0.0000000E+00	-6000.000	2.8275235E-08
0.0000000E+00	-5000.000	3.8069030E-07
0.0000000E+00	-4000.000	2.1086141E-06
0.0000000E+00	-3000.000	4.1301118E-06

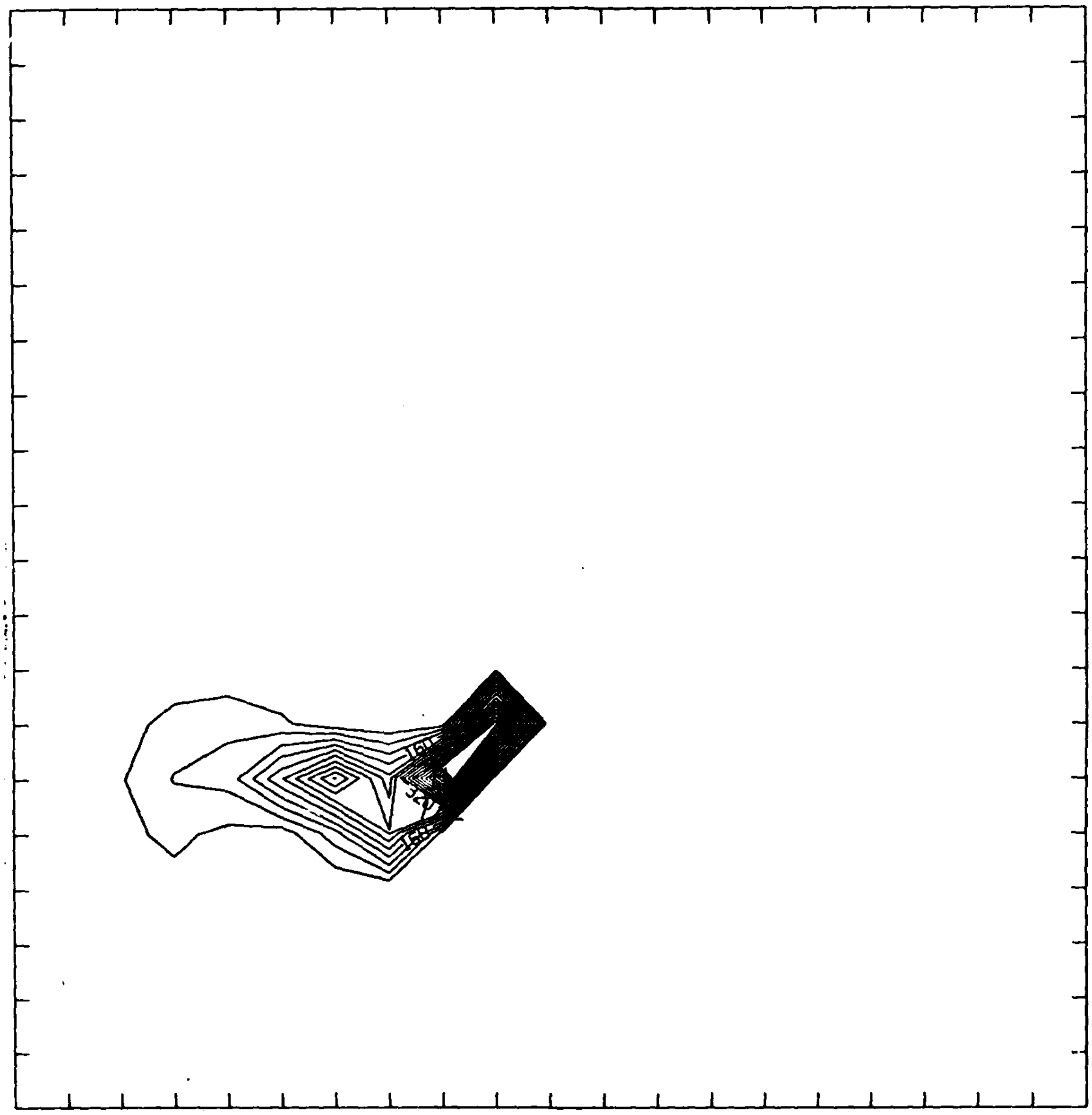
0.0000000E+00	-2000.000	4.2579782E-06
0.0000000E+00	-1000.000	1.9503220E-06
0.0000000E+00	0.0000000E+00	2.4262669E-03
1000.000	-5000.000	2.0946805E-08
1000.000	-4000.000	1.0375020E-07
1000.000	-3000.000	2.0892720E-07
1000.000	-2000.000	1.7105516E-07
1000.000	-1000.000	5.1520832E-08
HOUR NO.	1	
-10000.00	-10000.00	5.5788860E-07
-10000.00	-9000.000	2.8156639E-06
-10000.00	-8000.000	8.0341515E-06
-10000.00	-7000.000	1.3545313E-05
-10000.00	-6000.000	1.5679681E-05
-10000.00	-5000.000	1.2036261E-05
-10000.00	-4000.000	6.1953192E-06
-10000.00	-3000.000	1.7575444E-06
-10000.00	-2000.000	1.8287656E-07
-9000.000	-10000.00	2.0645091E-06
-9000.000	-9000.000	9.4914267E-06
-9000.000	-8000.000	2.8234637E-05
-9000.000	-7000.000	5.6195349E-05
-9000.000	-6000.000	7.7877237E-05
-9000.000	-5000.000	8.3980260E-05
-9000.000	-4000.000	6.5899490E-05
-9000.000	-3000.000	2.5836225E-05
-9000.000	-2000.000	3.3700903E-06
-9000.000	-1000.000	1.1056100E-07
-8000.000	-10000.00	4.1697040E-06
-8000.000	-9000.000	1.9427493E-05
-8000.000	-8000.000	5.9357877E-05
-8000.000	-7000.000	1.2692247E-04
-8000.000	-6000.000	2.1000027E-04
-8000.000	-5000.000	3.0859446E-04
-8000.000	-4000.000	3.6349453E-04
-8000.000	-3000.000	1.9253913E-04
-8000.000	-2000.000	2.9846569E-05
-8000.000	-1000.000	9.9646559E-07
-7000.000	-10000.00	4.6198434E-06
-7000.000	-9000.000	2.3851180E-05
-7000.000	-8000.000	7.3673749E-05
-7000.000	-7000.000	1.5124744E-04
-7000.000	-6000.000	2.7167599E-04
-7000.000	-5000.000	4.8636139E-04
-7000.000	-4000.000	8.2529243E-04
-7000.000	-3000.000	5.9163861E-04
-7000.000	-2000.000	9.0158734E-05
-7000.000	-1000.000	2.8070544E-06
-6000.000	-10000.00	3.0926890E-06
-6000.000	-9000.000	1.5905884E-05
-6000.000	-8000.000	4.8556074E-05
-6000.000	-7000.000	9.6805641E-05
-6000.000	-6000.000	1.6303528E-04
-6000.000	-5000.000	2.9571843E-04
-6000.000	-4000.000	9.4281795E-04
-6000.000	-3000.000	7.3076697E-04
-6000.000	-2000.000	8.7169254E-05
-6000.000	-1000.000	2.0855348E-06
-5000.000	-10000.00	1.2416035E-06
-5000.000	-9000.000	5.7770721E-06

-5000.000	-8000.000	1.7117491E-05
-5000.000	-7000.000	3.1588708E-05
-5000.000	-6000.000	4.4729994E-05
-5000.000	-5000.000	1.6049319E-04
-5000.000	-4000.000	2.3481811E-03
-5000.000	-3000.000	4.8578763E-04
-5000.000	-2000.000	2.0307807E-05
-5000.000	-1000.000	3.1001C22E-07
-4000.000	-10000.00	2.5067547E-07
-4000.000	-9000.000	1.2676365E-06
-4000.000	-8000.000	3.6346867E-06
-4000.000	-7000.000	6.1686324E-06
-4000.000	-6000.000	7.2026774E-06
-4000.000	-5000.000	1.0435894E-03
-4000.000	-4000.000	4.0392103E-03
-4000.000	-3000.000	6.8359324E-05
-4000.000	-2000.000	6.5204460E-07
-3000.000	-10000.00	3.3925232E-08
-3000.000	-9000.000	1.5204233E-07
-3000.000	-8000.000	4.2758114E-07
-3000.000	-7000.000	6.8059296E-07
-3000.000	-6000.000	6.8683090E-07
-3000.000	-5000.000	2.8345170E-03
-3000.000	-4000.000	2.1578320E-03
-3000.000	-3000.000	1.3792969E-08
-2000.000	-9000.000	1.1292729E-08
-2000.000	-8000.000	3.0696821E-08
-2000.000	-7000.000	4.5794284E-08
-2000.000	-6000.000	4.1436365E-08
-2000.000	-5000.000	2.2815168E-04
-2000.000	-4000.000	7.2240145E-03
-2000.000	-3000.000	3.7894868E-05
-1000.000	-4000.000	6.1226558E-05
-1000.000	-3000.000	5.8518853E-03
-1000.000	-2000.000	1.6241506E-04
0.0000000E+00	-1000.000	1.0502585E-05



CØNTØUR FRØM 0.00000E+00 TØ 0.24000E-02 CØNTØUR INTERVAL ØF 0.10000E-03 PT(3,3)= 0.00000E+00 LABELS SCALED BY 0.10000E+06

FIGURE 5(a) GRAPHICAL OUTPUTS FOR SAMPLE PROBLEM 2



CØNTOUR FRØM 0.00000E+00 TØ 0.72000E-02 CØNTØUR INTERVAL ØF 0.40000E-03 PT(3,3)= 0.59358E-04 LABELS SCALED BY 0.10000E+06

FIGURE 5(b) GRAPHICAL OUTPUTS FOR SAMPLE PROBLEM 2 (concluded)

END

FILMED

3-86

DTIC